



Numerical stability of a hybrid method for pricing options

Maya Briani, Lucia Caramellino, Giulia Terenzi, Antonino Zanette

► To cite this version:

Maya Briani, Lucia Caramellino, Giulia Terenzi, Antonino Zanette. Numerical stability of a hybrid method for pricing options. *International Journal of Theoretical and Applied Finance*, 2019, pp.1950036. 10.1142/S0219024919500365 . hal-02380723

HAL Id: hal-02380723

<https://hal.science/hal-02380723>

Submitted on 28 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numerical stability of a hybrid method for pricing options

MAYA BRIANI^{*}
LUCIA CARAMELLINO[†]
GIULIA TERENCE[‡]
ANTONINO ZANETTE[§]

Abstract

We develop and study stability properties of a hybrid approximation of functionals of the Bates jump model with stochastic interest rate that uses a tree method in the direction of the volatility and the interest rate and a finite-difference approach in order to handle the underlying asset price process. We also propose hybrid simulations for the model, following a binomial tree in the direction of both the volatility and the interest rate, and a space-continuous approximation for the underlying asset price process coming from a Euler-Maruyama type scheme. We test our numerical schemes by computing European and American option prices.

Keywords: stochastic volatility; jump-diffusion process; European and American options; tree methods; finite-difference; numerical stability.

2000 MSC: 91G10, 60H30, 65C20.

1 Introduction

Following the work in [8, 9], we further develop and study the hybrid tree/finite-difference approach and the hybrid Monte Carlo technique in order to numerically evaluate option prices. We concern here with the theoretical study of the stability of the numerical scheme for both European and American options. Also, in this paper we stress the model (and the associated numerical procedure) by considering the Bates model [7], possibly coupled with a stochastic interest rate following the Vasicek dynamics [41], and we call the full model as Bates-Hull-White.

The option pricing tree/finite-difference approach we deal with, derives from applying an efficient simple (recombining binomial) tree method in the direction of the volatility and the interest rate components, whereas the asset price component is locally treated by means of a one-dimensional partial integro-differential equation (PIDE), to which a finite-difference scheme is applied. Here, the numerical treatment of the nonlocal term coming from the jumps involves implicit-explicit techniques, as well as numerical quadratures. The procedure applies to other rather general stochastic volatility models (see [10]) or for problems in insurance (see [23]). We concentrate here our attention to the

^{*}Istituto per le Applicazioni del Calcolo, CNR Roma - m.briani@iac.cnr.it

[†]Dipartimento di Matematica, Università di Roma Tor Vergata, and INdAM-GNAMPA - caramell@mat.uniroma2.it

[‡]Dipartimento di Matematica, Università di Roma Tor Vergata, and LAMA, Université Paris-Est - terenzi@mat.uniroma2.it

[§]Dipartimento di Scienze Economiche e Statistiche, Università di Udine - antonino.zanette@uniud.it

Bates-Hull-White model and we give results on the numerical stability. Let us mention that, to this purpose, we never require the validity of the Feller condition for the Cox-Ingersoll-Ross (CIR) dynamics [18] of the volatility process.

In the case of plain vanilla European options, Fourier inversion methods [15] lead to closed-form formulas to compute the price under the Bates model. For American options, numerical methods are typically based on the use of the dynamic programming principle to which one applies either deterministic schemes for solutions of PIDEs from numerical analysis and/or from tree methods or Monte Carlo techniques. Our method is a mixture between a tree approach and a deterministic numerical approach, and is particularly tailored for the use of the backward dynamic programming principle.

Let us recall that tree methods for the Heston model have been already proposed in the literature. For example, Akyildirim, Dolinsky and Soner [1] have recently provided a four tuple discrete Markov approximation which can be generalized to other stochastic volatility models with a factor equation. Lo, Nguyen and Skindilis [32] have proposed a Markov chain approximation based on a trinomial tree, again adaptable to other stochastic volatility models. Vellekoop and Nieuwenhuis [40] have introduced a binomial tree built from the full truncation scheme of Lord, Koekkoek and Dijk [34]. Other tree approaches for the Heston model are available (see e.g. the references quoted in [40]). Generally, the Feller condition for the CIR volatility equation is required, either for theoretical purposes or for the numerical efficiency of the method.

Another tool is given by the discretization of PDEs. When the jumps are not considered, available references are recalled in [8, 9]. In the standard Bates model, the finite-difference methods for solving the 2-dimensional PIDE associated with the option pricing problems can be based on implicit, explicit or alternating direction implicit schemes. The implicit scheme requires to solve a dense sparse system at each time step. Toivanen [39] proposes a componentwise splitting method for pricing American options. In Ballestra and Cecere [10], the problem is handled by using an ad hoc pseudospectral method. Chiarella, Kang, Meyer and Ziogas [16] developed a method of lines algorithm for pricing and hedging American options again under the standard Bates dynamics. Itkin [28] has recently proposed a unified approach to handle PIDEs associated with Lévy's models of interest in Finance.

From the simulation point of view, the main problem consists in the treatment of the CIR dynamics for the volatility process. Several efficient and accurate methods have been developed specifically for the simulation of CIR paths, see, e.g., Alfonsi [2], Andersen [3], Lord, Koekkoek and Dijk [34] or Kahl and Jäckel [29]. We propose here a hybrid Monte Carlo technique: we couple the simulation of the approximating tree for the volatility and the interest rate components with a standard simulation of the underlying asset price, based on Brownian increments and a straightforward treatment of the jumps. In the case of American option, this is associated with the Longstaff and Schwartz algorithm [33], allowing to treat the dynamic programming principle. The numerical results are then compared with the Alfonsi's third-order simulation scheme.

The paper is organized as follows. In Section 2, we introduce the Bates-Hull-White model. In Section 3 we recall the tree procedure for the volatility and the interest rate pair (Section 3.1), we describe our discretization of the log-price process (Section 3.2) and the hybrid Monte Carlo simulations (Section 3.3). Section 4 is devoted to the hybrid tree/finite-difference method: we set the numerical scheme for the associated local PIDE problem (Section 4.1) and we apply it to the solution of the whole pricing scheme (Section 4.2). Section 5 is devoted to the analysis of the numerical stability of the resulting tree/finite-difference method. Section 6 refers to the practical use of our methods. Here, numerical results and comparisons are widely discussed.

2 The Bates-Hull-White model

The Bates model [7] is a stochastic volatility model with price jumps: the dynamics of the underlying asset price is driven by both a Heston stochastic volatility [26] and a compound Poisson jump process of the type originally introduced by Merton [35]. We allow the interest rate to follow a stochastic model, which we assume to be a generalized Ornstein-Uhlenbeck (hereafter OU) process. More precisely, the dynamics under the risk neutral measure of the share price S , the volatility process V and the interest rate r , are given by the following jump-diffusion model:

$$\begin{aligned}\frac{dS_t}{S_{t-}} &= (r_t - \eta)dt + \sqrt{V_t} dZ_t^S + dH_t, \\ dV_t &= \kappa_V(\theta_V - V_t)dt + \sigma_V \sqrt{V_t} dZ_t^V, \\ dr_t &= \kappa_r(\theta_r(t) - r_t)dt + \sigma_r dZ_t^r,\end{aligned}\tag{2.1}$$

where η denotes the continuous dividend rate, $S_0, V_0, r_0 > 0$, Z^S , Z^V and Z^r are correlated Brownian motions and H is a compound Poisson process with intensity λ and i.i.d. jumps $\{J_k\}_k$, that is

$$H_t = \sum_{k=1}^{K_t} J_k,\tag{2.2}$$

K denoting a Poisson process with intensity λ . We assume that the Poisson process K , the jump amplitudes $\{J_k\}_k$ and the 3-dimensional correlated Brownian motion (Z^S, Z^V, Z^r) are independent. As suggested by Grzelak and Oosterlee in [24], the significant correlations are between the noises governing the pairs (S, V) and (S, r) . So, as done in [9], we assume that the couple (Z^V, Z^r) is a standard Brownian motion in \mathbb{R}^2 and Z^S is a Brownian motion in \mathbb{R} which is correlated both with Z^V and Z^r :

$$d\langle Z^S, Z^V \rangle_t = \rho_1 dt \quad \text{and} \quad d\langle Z^S, Z^r \rangle_t = \rho_2 dt.$$

We recall that the volatility process V follows a CIR dynamics with mean reversion rate κ_V , long run variance θ_V and σ_V denotes the vol-vol (volatility of the volatility). We assume that $\theta_V, \kappa_V, \sigma_V > 0$ and we stress that we never require in this paper that the CIR process satisfies the Feller condition $2\kappa_V\theta_V \geq \sigma_V^2$, ensuring that the process V never hits 0. So, we allow the volatility V to reach 0. The interest rate r_t is described by a generalized OU process, in particular θ_r is time-dependent but deterministic and fits the zero-coupon bond market values, for details see [13]. As already done in [27], we write the process r as follows:

$$r_t = \sigma_r X_t + \varphi_t\tag{2.3}$$

where

$$X_t = -\kappa_r \int_0^t X_s ds + Z_t^r \quad \text{and} \quad \varphi_t = r_0 e^{-\kappa_r t} + \kappa_r \int_0^t \theta_r(s) e^{-\kappa_r(t-s)} ds.\tag{2.4}$$

From now on we set

$$Z^V = W^1, \quad Z^r = W^2, \quad Z^S = \rho_1 W^1 + \rho_2 W^2 + \rho_3 W^3,$$

where $W = (W^1, W^2, W^3)$ is a standard Brownian motion in \mathbb{R}^3 and the correlation parameter ρ_3 is given by

$$\rho_3 = \sqrt{1 - \rho_1^2 - \rho_2^2}, \quad \rho_1^2 + \rho_2^2 \leq 1.$$

By passing to the logarithm $Y = \ln S$ in the first component, by taking into account the above mentioned correlations and by considering the process X as in (2.3)-(2.4), we get the triple (Y, V, X) given by

$$\begin{aligned} dY_t &= \mu_Y(V_t, X_t, t)dt + \sqrt{V_t}(\rho_1 dW_t^1 + \rho_2 dW_t^2 + \rho_3 dW_t^3) + dN_t, \quad Y_0 = \ln S_0 \in \mathbb{R}, \\ dV_t &= \mu_V(V_t)dt + \sigma_V \sqrt{V_t} dW_t^1, \quad V_0 > 0, \\ dX_t &= \mu_X(X_t)dt + dW_t^2, \quad X_0 = 0, \end{aligned} \quad (2.5)$$

where

$$\mu_Y(v, x, t) = \sigma_r x + \varphi_t - \eta - \frac{1}{2}v, \quad (2.6)$$

$$\mu_V(v) = \kappa_V(\theta_V - v), \quad (2.7)$$

$$\mu_X(x) = -\kappa_r x, \quad (2.8)$$

and N_t is the compound Poisson process written through the Poisson process K , with intensity λ , and the i.i.d. jumps $\{\log(1 + J_k)\}_k$, that is

$$N_t = \sum_{k=1}^{K_t} \log(1 + J_k),$$

Recall that K , the jump amplitudes $\{\log(1 + J_k)\}_k$ and the 3-dimensional standard Brownian motion (W^1, W^2, W^3) are all independent. We also recall that the Lévy measure associated with N is given by

$$\nu(dx) = \lambda \mathbb{P}(\log(1 + J_1) \in dx),$$

and whenever $\log(1 + J_1)$ is absolutely continuous then ν has a density as well:

$$\nu(dx) = \nu(x)dx = \lambda p_{\log(1+J_1)}(x)dx, \quad (2.9)$$

$p_{\log(1+J_1)}$ denoting the probability density function of $\log(1 + J_1)$. For example, in the Merton model [35] it is assumed that $\log(1 + J_1)$ has a normal distribution - this is the choice we will do in our numerical experiments, as done in Chiarella *et al.* [16]. But other jump-amplitude measures can be selected. For instance, in the Kou model [30] the law of $\log(1 + J_1)$ is a mixture of exponential laws:

$$p_{\log(1+J_1)}(x) = p\lambda_+ e^{-\lambda_+ x} 1_{\{x>0\}} + (1-p)\lambda_- e^{\lambda_- x} 1_{\{x<0\}},$$

1_A denoting the indicator function of A . Here, the parameters $\lambda_{\pm} > 0$ control the decrease of the distribution tails of negative and positive jumps respectively, and p is the probability of a positive jump.

We pass to the transformation $Y = \ln S$. If $\Psi(Y)$ denotes the payoff written on the log-price, the option price $P = P(t, y, v, x)$ is given by

$$\begin{aligned} \text{European price:} \quad P(t, y, v, x) &= \mathbb{E} \left(e^{-\int_t^T (\sigma_r X_s^{t,x} + \varphi_s) ds} \Psi(Y_T^{t,y,v,x}) \right), \\ \text{American price:} \quad P(t, y, v, x) &= \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E} \left(e^{-\int_t^\tau (\sigma_r X_s^{t,x} + \varphi_s) ds} \Psi(Y_\tau^{t,y,v,x}) \right), \end{aligned} \quad (2.10)$$

where $\mathcal{T}_{t,T}$ denotes the set of all stopping times taking values on $[t, T]$. Hereafter, $(Y^{t,y,v,x}, V^{t,v}, X^{t,x})$ denotes the solution of the jump-diffusion dynamic (2.5) starting at time t in the point (y, v, x) .

3 The dicretized process

We first set up the discretization of the triple (Y, V, X) we will take into account.

3.1 The 2-dimensional tree for (V, X)

We consider an approximation for the pair (V, X) in (2.5) on the time-interval $[0, T]$ by means of a 2-dimensional computationally simple tree. This means that we construct a Markov chain running over a 2-dimensional recombining bivariate lattice and, at each time-step, both components of the Markov chain can jump only upwards or downwards. We consider the “multiple-jumps” approach by Nelson and Ramaswamy [36], extensively developed for the CIR process in [5]. We give here the main ideas in order to set-up the whole algorithm. We start by considering a discretization of the time-interval $[0, T]$ in N subintervals $[nh, (n+1)h]$, $n = 0, 1, \dots, N$, with $h = T/N$.

For $n = 0, 1, \dots, N$, consider the lattice for V and X defined by

$$\mathcal{V}_n = \{v_k^n\}_{k=0,1,\dots,n} \quad \text{with} \quad v_k^n = \left(\sqrt{V_0} + \frac{\sigma_V}{2}(2k-n)\sqrt{h} \right)^2 1_{\{\sqrt{V_0} + \frac{\sigma_V}{2}(2k-n)\sqrt{h} > 0\}}, \quad (3.1)$$

$$\mathcal{X}_n = \{x_j^n\}_{j=0,1,\dots,n} \quad \text{with} \quad x_j^n = (2j-n)\sqrt{h}, \quad (3.2)$$

respectively. Notice that $v_0^0 = V_0$ and $x_0^0 = 0 = X_0$. For each fixed $v_k^n \in \mathcal{V}_n$ and $x_j^n \in \mathcal{X}_n$, we denote the “up” and “down” jump by $v_{k_u(n,k)}^{n+1}$ and $v_{k_d(n,k)}^{n+1}$ and by $x_{j_u(n,j)}^{n+1}$ and $x_{j_d(n,j)}^{n+1}$. By applying the “multiple jump approach”, the jump-indexes $k_u(n, k)$, $k_d(n, k)$, $j_u(n, j)$, $j_d(n, j)$ are defined as

$$k_u(n, k) = \min\{k^* : k+1 \leq k^* \leq n+1 \text{ and } v_k^n + \mu_V(v_k^n)h \leq v_{k^*}^{n+1}\}, \quad (3.3)$$

$$k_d(n, k) = \max\{k^* : 0 \leq k^* \leq k \text{ and } v_k^n + \mu_V(v_k^n)h \geq v_{k^*}^{n+1}\}, \quad (3.4)$$

$$j_u(n, j) = \min\{j^* : j+1 \leq j^* \leq n+1 \text{ and } x_j^n + \mu_X(x_j^n)h \leq x_{j^*}^{n+1}\}, \quad (3.5)$$

$$j_d(n, j) = \max\{j^* : 0 \leq j^* \leq j \text{ and } x_j^n + \mu_X(x_j^n)h \geq x_{j^*}^{n+1}\}, \quad (3.6)$$

where μ_V and μ_X is the drift of V and X respectively (see (2.7) and (2.8)), with the understanding $k_u(n, k) = n+1$, respectively $k_d(n, k) = 0$, if the set in the r.h.s. of (3.3), respectively (3.4), is empty, and similarly for j_u and j_d .

The transition probabilities for V are defined as follows: starting from the node (n, k) the probability that the process jumps to $k_u(n, k)$ and $k_d(n, k)$ at time-step $n+1$ are set as

$$p_u^V(n, k) = 0 \vee \frac{\mu_V(v_k^n)h + v_k^n - v_{k_d(n,k)}^{n+1}}{v_{k_u(n,k)}^{n+1} - v_{k_d(n,k)}^{n+1}} \wedge 1 \quad \text{and} \quad p_d^V(n, k) = 1 - p_u^V(n, k) \quad (3.7)$$

respectively. A similar definition is set for the component X : starting from (n, j) , the probability that the process jumps to $j_u(n, j)$ and $j_d(n, j)$ at time-step $n+1$ are

$$p_u^X(n, j) = 0 \vee \frac{\mu_X(x_j^n)h + x_j^n - x_{j_d(n,j)}^{n+1}}{x_{j_u(n,j)}^{n+1} - x_{j_d(n,j)}^{n+1}} \wedge 1 \quad \text{and} \quad p_d^X(n, j) = 1 - p_u^X(n, j) \quad (3.8)$$

respectively.

The tree procedure for the pair (V, X) is obtained by joining the trees built for V and for X . Namely, for $n = 0, 1, \dots, N$, consider the lattice

$$\mathcal{V}_n \times \mathcal{X}_n = \{(v_k^n, x_j^n)\}_{k,j=0,1,\dots,n}. \quad (3.9)$$

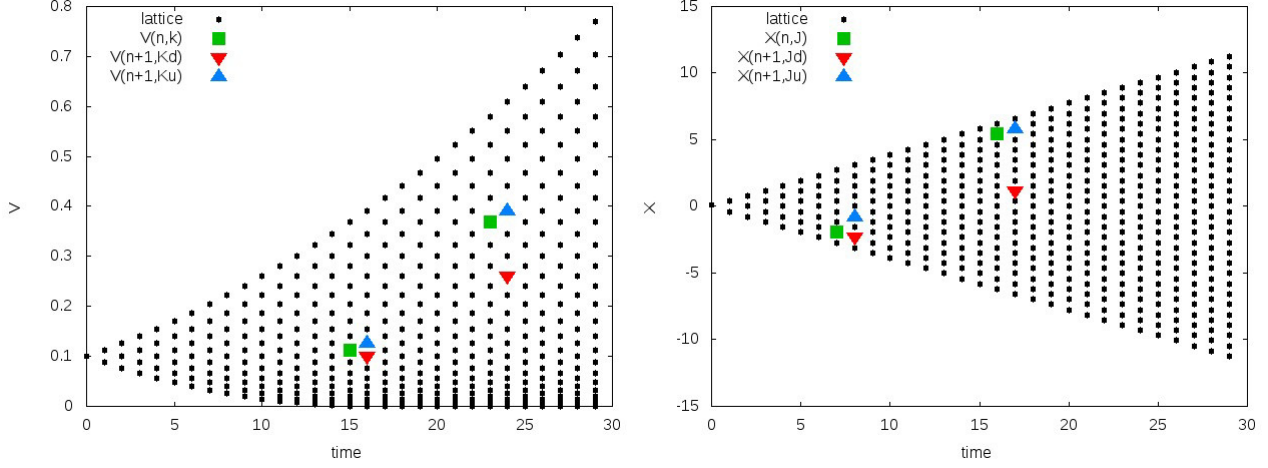


Figure 1: The tree for the process V (left) and for X (right). The figure shows possible instances of the up and down jumps.

Starting from the node (n, k, j) , which corresponds to the position $(v_k^n, x_j^n) \in \mathcal{V}_n \times \mathcal{X}_n$, we define the four possible jumps by means of the following four nodes at time $n + 1$:

$$\begin{aligned}
 (n+1, k_u(n, k), j_u(n, j)) & \quad \text{with probability} \quad p_{uu}(n, k, j) = p_u^V(n, k)p_u^X(n, j), \\
 (n+1, k_u(n, k), j_d(n, j)) & \quad \text{with probability} \quad p_{ud}(n, k, j) = p_u^V(n, k)p_d^X(n, j), \\
 (n+1, k_d(n, k), j_u(n, j)) & \quad \text{with probability} \quad p_{du}(n, k, j) = p_d^V(n, k)p_u^X(n, j), \\
 (n+1, k_d(n, k), j_d(n, j)) & \quad \text{with probability} \quad p_{dd}(n, k, j) = p_d^V(n, k)p_d^X(n, j),
 \end{aligned} \tag{3.10}$$

where, for $a \in \{u, d\}$, the above nodes $k_a(n, k)$, $j_a(n, j)$ and the above probabilities $p_a^V(n, k)$, $p_a^X(n, j)$ are defined in (3.3)-(3.4), (3.5)-(3.6), (3.7) and (3.8). The factorization of the jump probabilities in (3.10) follows from the orthogonality property of the noises driving the two processes. This procedure gives rise to a Markov chain $(\hat{V}_n^h, \hat{X}_n^h)_{n=0, \dots, N}$ that weakly converges on the path space, as $h \rightarrow 0$, to the diffusion process $(V_t, X_t)_{t \in [0, T]}$ solution to

$$\begin{aligned}
 dV_t &= \mu_V(V_t)dt + \sigma_V \sqrt{V_t} dW_t^1, \quad V_0 > 0, \\
 dX_t &= \mu_X(X_t)dt + dW_t^2, \quad X_0 = 0.
 \end{aligned}$$

This can be proved by using standard results (see e.g. the techniques in [36]) and the convergence of the chain approximating the volatility process proved in [5]. And this holds independently of the validity of the Feller condition $2\kappa_V\theta_V \geq \sigma_V^2$.

Details and remarks on the extension of this procedure to more general cases can be found in [9]. In particular, if the correlation between the Brownian motions driving (V, X) was not null, one could define the jump probabilities by matching the local cross-moment (see Remark 3.1 in [9]).

3.2 The approximation of the component Y

We describe here how we manage the Y -component in (2.5) by taking into account the tree procedure given for the pair (V, X) . We go back to (2.5): by isolating $\sqrt{V_t}dW_t^1$ in the second line and dW_t^2 in the third one, we obtain

$$dY_t = \mu(V_t, X_t, t)dt + \rho_3 \sqrt{V_t} dW_t^3 + \frac{\rho_1}{\sigma_V} dV_t + \rho_2 \sqrt{V_t} dX_t + dN_t \tag{3.11}$$

with

$$\begin{aligned}\mu(v, x, t) &= \mu_Y(v, x, t) - \frac{\rho_1}{\sigma_V} \mu_V(v) - \rho_2 \sqrt{v} \mu_X(x) \\ &= \sigma_r x + \varphi_t - \eta - \frac{1}{2} v - \frac{\rho_1}{\sigma_V} \kappa_V(\theta_V - v) + \rho_2 \kappa_r x \sqrt{v}\end{aligned}\quad (3.12)$$

(μ_Y , μ_V and μ_X are defined in (2.6), (2.7) and (2.8) respectively). To numerically solve (3.11), we mainly use the fact that the noises W^3 and N are independent of the processes V and X . So, we first take the approximating tree $(\hat{V}_n^h, \hat{X}_n^h)_{n=0,1,\dots,N-1}$ discussed in Section 3.1 and we set $(\bar{V}_t^h, \bar{X}_t^h)_{t \in [0,T]} = (\hat{V}_{[t/h]}^h, \hat{X}_{[t/h]}^h)_{t \in [0,T]}$ the associated time-continuous càdlàg approximating process for (V, X) . Then, we insert the discretization (\bar{V}^h, \bar{X}^h) for (V, X) in the coefficients of (3.11). Therefore, the final process \bar{Y}^h approximating Y is set as follows: $\bar{Y}_0^h = Y_0$ and for $t \in (nh, (n+1)h]$ with $n = 0, 1, \dots, N-1$

$$\begin{aligned}\bar{Y}_t^h &= \bar{Y}_{nh}^h + \mu(\bar{V}_{nh}^h, \bar{X}_{nh}^h, nh)(t - nh) + \rho_3 \sqrt{\bar{V}_t^h} (W_t^3 - W_{nh}^3) \\ &\quad + \frac{\rho_1}{\sigma_V} (\bar{V}_t^h - \bar{V}_{nh}^h) + \rho_2 \sqrt{\bar{V}_t^h} (\bar{X}_t^h - \bar{X}_{nh}^h) + (N_t - N_{nh}).\end{aligned}\quad (3.13)$$

3.3 The Monte Carlo approach

Let us show how one can simulate a single path by using the tree approximation (3.9) for the couple (V, X) and the Euler scheme (3.13) for the Y -component.

Let $(\hat{Y}_n)_{n=0,1,\dots,N}$ be the sequence approximating Y at times nh , $n = 0, 1, \dots, N$, by means of the scheme in (3.13): $\hat{Y}_0^h = Y_0$ and for $t \in [nh, (n+1)h]$ with $n = 0, 1, \dots, N-1$ then

$$\begin{aligned}\hat{Y}_{n+1}^h &= \hat{Y}_n^h + \mu(\hat{V}_n^h, \hat{X}_n^h, nh)h + \rho_3 \sqrt{h \hat{V}_n^h} \Delta_{n+1} \\ &\quad + \frac{\rho_1}{\sigma_V} (\hat{V}_{n+1}^h - \hat{V}_n^h) + \rho_2 \sqrt{\hat{V}_n^h} (\hat{X}_{n+1}^h - \hat{X}_n^h) + (N_{(n+1)h} - N_{nh}),\end{aligned}$$

where μ is defined in (3.12) and $\Delta_1, \dots, \Delta_N$ denote i.i.d. standard normal r.v.'s, independent of the noise driving the chain (\hat{V}, \hat{X}) . The simulation of $N_{(n+1)h} - N_{nh}$ is straightforward: one first generates a Poisson r.v. K_h^{n+1} of parameter λh and if $K_h^{n+1} > 0$ then also the log-amplitudes $\log(1 + J_k^{n+1})$ for $k = 1, \dots, K_h^{n+1}$ are simulated. Then, the observed jump of the compound Poisson process is written as the sum of the simulated log-amplitudes, so that

$$\begin{aligned}\hat{Y}_{n+1}^h &= \hat{Y}_n^h + \mu(\hat{V}_n^h, \hat{X}_n^h, nh)h + \rho_3 \sqrt{h \hat{V}_n^h} \Delta_{n+1} \\ &\quad + \frac{\rho_1}{\sigma_V} (\hat{V}_{n+1}^h - \hat{V}_n^h) + \rho_2 \sqrt{\hat{V}_n^h} (\hat{X}_{n+1}^h - \hat{X}_n^h) + \sum_{k=1}^{K_h^{n+1}} \log(1 + J_k^{n+1}),\end{aligned}\quad (3.14)$$

in which the last sum is set equal to 0 if $K_h^{n+1} = 0$.

The above simulation scheme is plain: at each time step $n \geq 1$, one let the pair (V, X) evolve on the tree and simulate the process Y by using (3.14). We will refer to this procedure as *hybrid Monte Carlo algorithm*, the word “hybrid” being related to the fact that two different noise sources are considered: we simulate a continuous process in space (the component Y) starting from a discrete process in space (the tree for (V, X)).

The simulations just described will be used in next Section 6 in order to set-up a Monte Carlo procedure for the computation of the option price function (2.10). In the case of American options, the simulations are coupled with the Monte Carlo algorithm by Longstaff and Schwartz in [33].

4 The hybrid tree/finite difference approach

The option price in (2.10) is typically computed by means of the standard backward dynamic programming algorithm. So, consider a discretization of the time interval $[0, T]$ into N subintervals of length $h = T/N$. Then the price $P(0, Y_0, V_0, X_0)$ is numerically approximated through the quantity $P_h(0, Y_0, V_0, X_0)$ backwardly given by

$$\begin{cases} P_h(T, y, v, x) = \Psi(y) & \text{and as } n = N - 1, \dots, 0, \\ P_h(nh, y, v, x) = \max \left\{ \widehat{\Psi}(y), e^{-(\sigma_r x + \varphi_{nh})h} \mathbb{E} \left(P_h((n+1)h, Y_{(n+1)h}^{nh, y, v, x}, V_{(n+1)h}^{nh, v}, X_{(n+1)h}^{nh, x}) \right) \right\}, \end{cases} \quad (4.1)$$

for $(y, v, x) \in \mathbb{R} \times \mathbb{R}_+ \times \mathbb{R}$, in which

$$\widehat{\Psi}(y) = \begin{cases} 0 & \text{in the European case,} \\ \Psi(y) & \text{in the American case.} \end{cases}$$

So, what is needed is a good approximation of the expectations appearing in the above dynamic programming principle. This is what we first deal with, starting from the discretized process $(\bar{Y}^h, \bar{V}^h, \bar{X}^h)$ introduced in Section 3.

4.1 The local 1-dimensional partial integro-differential equation

Let \bar{Y}^h denote the process in (3.13). If we set

$$\bar{Z}_t^h = \bar{Y}_t^h - \frac{\rho_1}{\sigma_V}(\bar{V}_t^h - \bar{V}_{nh}^h) - \rho_2 \sqrt{\bar{V}_{nh}^h}(\bar{X}_t^h - \bar{X}_{nh}^h), \quad t \in [nh, (n+1)h] \quad (4.2)$$

then we have (μ being given in (3.12))

$$\begin{aligned} d\bar{Z}_t^h &= \mu(\bar{V}_{nh}^h, \bar{X}_{nh}^h, nh)dt + \rho_3 \sqrt{\bar{V}_{nh}^h} dW_t^3 + dN_t \quad t \in (nh, (n+1)h], \\ \bar{Z}_{nh}^h &= \bar{Y}_{nh}^h, \end{aligned} \quad (4.3)$$

that is, \bar{Z}^h solves a jump-diffusion stochastic equation with constant coefficients and at time nh it starts from \bar{Y}_{nh}^h . Take now a function f : we are interested in computing

$$\mathbb{E}(f(Y_{(n+1)h}) \mid Y_{nh} = y, V_{nh} = v, X_{nh} = x).$$

We actually need a function f of the whole variables (y, v, x) but at the present moment the variable y is the most important one, we will see later on that the introduction of (v, x) is straightforward. So, we numerically compute the above expectation by means of the one done on the approximating processes, that is,

$$\begin{aligned} &\mathbb{E}(f(\bar{Y}_{(n+1)h}^h) \mid \bar{Y}_{nh}^h = y, \bar{V}_{nh}^h = v, \bar{X}_{nh}^h = x) \\ &= \mathbb{E}(f(\bar{Z}_{(n+1)h}^h + \frac{\rho_1}{\sigma_V}(\bar{V}_{(n+1)h}^h - \bar{V}_{nh}^h) + \rho_2 \sqrt{\bar{V}_{nh}^h}(\bar{X}_{(n+1)h}^h - \bar{X}_{nh}^h)) \mid \bar{Z}_{nh}^h = y, \bar{V}_{nh}^h = v, \bar{X}_{nh}^h = x), \end{aligned}$$

in which we have used the process \bar{Z}^h in (4.2). Since (\bar{V}^h, \bar{X}^h) is independent of the Brownian noise W^3 and on the compound Poisson process N driving \bar{Z}^h in (4.3), we have the following: we set

$$\Psi_f(\zeta; y, v, x) = \mathbb{E}(f(\bar{Z}_{(n+1)h}^h + \zeta) \mid \bar{Z}_{nh}^h = y, \bar{V}_{nh}^h = v, \bar{X}_{nh}^h = x) \quad (4.4)$$

and we can write

$$\begin{aligned} \mathbb{E}(f(\bar{Y}_{(n+1)h}^h) \mid \bar{Y}_{nh}^h = y, \bar{V}_{nh}^h = v, \bar{X}_{nh}^h = x) \\ = \mathbb{E}\left(\Psi_f\left(\frac{\rho_1}{\sigma_V}(\bar{V}_{(n+1)h}^h - \bar{V}_{nh}^h) + \rho_2\sqrt{v}(\bar{X}_{(n+1)h}^h - \bar{X}_{nh}^h); y, v, x\right) \mid \bar{V}_{nh}^h = v, \bar{X}_{nh}^h = x\right). \end{aligned} \quad (4.5)$$

Now, in order to compute the quantity $\Psi_f(\zeta)$ in (4.4), we consider a generic function g and set

$$u(t, y; v, x) = \mathbb{E}(g(\bar{Z}_{(n+1)h}^h) \mid \bar{Z}_t^h = y, \bar{V}_t^h = v, \bar{X}_t^h = x), \quad t \in [nh, (n+1)h].$$

By (4.3) and the Feynman-Kac representation formula we can state that, for every fixed $x \in \mathbb{R}$ and $v \geq 0$, the function $(t, y) \mapsto u(t, y; v, x)$ is the solution to

$$\begin{cases} \partial_t u(t, y; v, x) + L^{(v, x)} u(t, y; v, x) = 0 & y \in \mathbb{R}, t \in [nh, (n+1)h), \\ u((n+1)h, y; v, x) = g(y) & y \in \mathbb{R}, \end{cases} \quad (4.6)$$

where $L^{(v, x)}$ is the integro-differential operator

$$\begin{aligned} L^{(v, x)} u(t, y; v, x) = & \mu(v, x) \partial_y u(t, y; v, x) + \frac{1}{2} \rho_3^2 v \partial_{yy}^2 u(t, y; v, x) \\ & + \int_{-\infty}^{+\infty} [u(t, y + \xi; v, x) - u(t, y; v, x)] \nu(\xi) d\xi, \end{aligned} \quad (4.7)$$

where μ is given in (3.12) and ν is the Lévy measure associated with the compound Poisson process N , see (2.9). We are assuming here that the Lévy measure is absolutely continuous (in practice, we use a Gaussian density), but it is clear that the procedure we are going to describe can be straightforwardly extended to other cases.

4.1.1 Finite-difference and numerical quadrature

In order to numerically compute the solution to the PIDE (4.6) at time nh , we generalize the approach already developed in [8, 9]: we apply a one-step finite-difference algorithm to the differential part of the problem coupled now with a quadrature rule to approximate the integral term.

We start by fixing an infinite grid on the y -axis $\mathcal{Y} = \{y_i = Y_0 + i\Delta y\}_{i \in \mathbb{Z}}$, with $\Delta y = y_i - y_{i-1}$, $i \in \mathbb{Z}$. For fixed n and given $x \in \mathbb{R}$ and $v \geq 0$, we set $u_i^n = u(nh, y_i; v, x)$ the discrete solution of (4.6) at time nh on the point y_i of the grid \mathcal{Y} – for simplicity of notations, in the sequel we do not stress in u_i^n the dependence on (v, x) .

First of all, to numerically compute the integral term in (4.7) we need to truncate the infinite integral domain to a bounded interval \mathcal{I} , to be taken large enough in order that

$$\int_{\mathcal{I}} \nu(\xi) d\xi \approx \lambda. \quad (4.8)$$

In terms of the process, this corresponds to truncate the large jumps. We assume that the tails of ν rapidly decrease – this is not really restrictive since applied models typically require that the tails of ν decrease exponentially. Hence, we take $R \in \mathbb{N}$ large enough, set $\mathcal{I} = [-R\Delta y, +R\Delta y]$ and apply to (4.8) the trapezoidal rule on the grid \mathcal{Y} with the same step Δy previously defined. Then, for $\xi_l = l\Delta y$, $l = -R, \dots, R$, we have

$$\int_{-R\Delta y}^{+R\Delta y} [u(t, y + \xi) - u(t, y)] \nu(\xi) d\xi \approx \Delta y \sum_{l=-R}^R (u(t, y + \xi_l) - u(t, y)) \nu(\xi_l). \quad (4.9)$$

We notice that $y_i + \xi_l = Y_0 + (i+l)\Delta y \in \mathcal{Y}$, so the values $u(t, y_i + \xi_l)$ are well defined on the numerical grid \mathcal{Y} for any i, l . These are technical settings and can be modified and calibrated for different Lévy measures ν .

But in practice one cannot solve the PIDE problem over the whole real line. So, we have to choose artificial bounds and impose numerical boundary conditions. We take a positive integer $M > 0$ and we define a finite grid $\mathcal{Y}_M = \{y_i = Y_0 + i\Delta y\}_{i \in \mathcal{J}_M}$, with $\mathcal{J}_M = \{-M, \dots, M\}$, and we assume that $M > R$. Notice that for $y = y_i \in \mathcal{Y}_M$ then the integral term in (4.9) splits into two parts: one part concerning nodes falling into the numerical domain \mathcal{Y}_M and another part concerning nodes falling out of \mathcal{Y}_M . As an example, at time $t = nh$ we have

$$\sum_{l=-R}^R u(nh, y_i + \xi_l) \nu(\xi_l) \approx \sum_{l=-R}^R u_{i+l}^n \nu(\xi_l) = \sum_{l: |l| \leq R, |i+l| \leq M} u_{i+l}^n \nu(\xi_l) + \sum_{l: |l| \leq R, |i+l| > M} \tilde{u}_{i+l}^n \nu(\xi_l),$$

where \tilde{u}^n stands for (unknown) values that fall out of the finite numerical domain \mathcal{Y}_M . This implies that we must choose some suitable artificial boundary conditions. In a financial context, in [19] it has been shown that a good choice for the boundary conditions is the payoff function. Although this is the choice we will do in our numerical experiments, for the sake of generality we assume here the boundary values outside \mathcal{Y}_M to be settled as $\tilde{u}_i^n = b(nh, y_i)$, where $b = b(t, y)$ is a fixed function defined in $[0, T] \times \mathbb{R}$.

Going back to the numerical scheme to solve the differential part of the equation (4.6), as already done in [9], we apply an implicit in time approximation. However, to avoid to solve at each time step a linear system with a dense matrix, the non-local integral term needs anyway an explicit in time approximation. We then obtain an implicit-explicit (hereafter IMEX) scheme as proposed in [19] and [11]. Notice that more sophisticated IMEX methods may be applied, see for instance [12, 38]. Let us stress that these techniques could be used in our framework, being more accurate but expensive.

As done in [9], to achieve greater precision we use the centered approximation for both first and second order derivatives in space. The discrete solution u^n at time nh is then computed in terms of the known value u^{n+1} at time $(n+1)h$ by solving the following discrete problem: for all $i \in \mathcal{J}_M$,

$$\frac{u_i^{n+1} - u_i^n}{h} + \tilde{\mu}_Y(v, x) \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta y} + \frac{1}{2} \rho_3^2 v \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta y^2} + \Delta y \sum_{l=-R}^R (u_{i+l}^{n+1} - u_i^{n+1}) \nu(\xi_l) = 0. \quad (4.10)$$

We then get the solution $u^n = (u_{-M}^n, \dots, u_M^n)^T$ by solving the following linear system

$$A u^n = B u^{n+1} + d, \quad (4.11)$$

where $A = A(v, x)$ and B are $(2M+1) \times (2M+1)$ matrices and d is a $(2M+1)$ -dimensional boundary vector defined as follows.

1. The matrix A . From (4.10), we set A as the tridiagonal real matrix given by

$$A = \begin{pmatrix} 1+2\beta & -\alpha-\beta & & & \\ \alpha-\beta & 1+2\beta & -\alpha-\beta & & \\ & \ddots & \ddots & \ddots & \\ & & \alpha-\beta & 1+2\beta & -\alpha-\beta \\ & & & \alpha-\beta & 1+2\beta \end{pmatrix}, \quad (4.12)$$

with

$$\alpha = \frac{h}{2\Delta y} \mu(nh, v, x) \quad \text{and} \quad \beta = \frac{h}{2\Delta y^2} \rho_3^2 v, \quad (4.13)$$

μ being defined in (3.12). We stress on that at each time step n , the quantities v and x are constant and known values (defined by the tree procedure for (V, X)) and then α and β are constant parameters.

2. The matrix B . Again from (4.10), B is the $(2M+1) \times (2M+1)$ real matrix given by

$$B = I + h\Delta y \begin{pmatrix} \nu(0) - \Lambda & \nu(\Delta y) & \dots & \nu(R\Delta y) & 0 \\ \nu(-\Delta y) & \nu(0) - \Lambda & \nu(\Delta y) & \dots & \nu(R\Delta y) \\ & \ddots & \ddots & \ddots & \\ 0 & \nu(-R\Delta y) & \dots & \nu(-\Delta y) & \nu(0) - \Lambda \end{pmatrix}, \quad (4.14)$$

where I is the identity matrix and

$$\Lambda = \sum_{l=-R}^R \nu(\xi_l).$$

3. The boundary vector d . The vector $d \in \mathbb{R}^{2M+1}$ contains the numerical boundary values:

$$d = a_b^n + a_b^{n+1}, \quad (4.15)$$

with

$$a_b^n = ((\beta - \alpha)b_{-M-1}^n, 0, \dots, 0, (\beta + \alpha)b_{M+1}^n)^T \in \mathbb{R}^{2M+1}$$

and $a_b^{n+1} \in \mathbb{R}^{2M+1}$ is such that

$$(a_b^{n+1})_i = \begin{cases} h\Delta y \sum_{l=-R}^{-M-i-1} \nu(x_l) b_{i+l}^{n+1}, & \text{for } i = -M, \dots, -M + R - 1, \\ 0 & \text{for } i = -M + R, \dots, M - R, \\ h\Delta y \sum_{l=M-i+1}^R \nu(x_l) b_{i+l}^{n+1}, & \text{for } i = M - R + 1, \dots, M - 1, \end{cases}$$

where we have used the standard notation $b_i^n = b(nh, y_i)$, $i \in \mathcal{J}_M$.

In practice, we numerically solve the linear system (4.11) with an efficient algorithm (see next Remark 6.2). We notice here that a solution to (4.11) really exists because for $\beta \neq |\alpha|$, the matrix $A = A(v, x)$ is invertible (see e.g. Theorem 2.1 in [14]). Then, at time nh , for each fixed $v \geq 0$ and $x \in \mathbb{R}$, we approximate the solution $y \mapsto u(nh, y; v, x)$ of (4.6) on the points y_i 's of the grid in terms of the discrete solution $u^n = \{u_i^n\}_{i \in \mathcal{J}_M}$, which in turn is written in terms of the value $u^{n+1} = \{u_i^{n+1}\}_{i \in \mathcal{J}_M}$ at time $(n+1)h$. In other words, we set

$$u(nh, y_i; v, x) \approx u_i^n, \quad i \in \mathcal{J}_M, \quad \text{where } u^n = (u_i^n)_{i \in \mathcal{J}_M} \text{ solves (4.11)} \quad (4.16)$$

4.1.2 The final local finite-difference approximation

We are now ready to tackle our original problem: the computation of the function $\Psi_f(\zeta; y, v, x)$ in (4.4) allowing one to numerically compute the expectation in (4.5). So, at time step n , the pair (v, x) is chosen on the lattice $\mathcal{V}_n \times \mathcal{X}_n$: $v = v_k^n$, $x = x_j^n$ for $0 \leq k, j \leq n$. We call $A_{k,j}^n$ the matrix A in (4.12) when evaluated in (v_k^n, x_j^n) and d^n the boundary vector in (4.15) at time-step n . Then, (4.16) gives

$$\begin{aligned} \Psi_f(\zeta; y_i, v_k^n, x_j^n) &\simeq u_{i,k,j}^n, \quad \text{where } u_{i,k,j}^n = (u_{i,k,j}^n)_{i \in \mathcal{J}_M} \text{ solves the linear system} \\ A_{k,j}^n u_{i,k,j}^n &= Bf(y_i + \zeta) + d^n. \end{aligned}$$

Therefore, by taking the expectation w.r.t. the tree-jumps, the expectation in (4.5) is finally computed on $\mathcal{Y}_M \times \mathcal{V}_n \times \mathcal{X}_n$ by means of the above approximation:

$$\begin{aligned} & \mathbb{E}(f(\bar{Y}_{(n+1)h}^h) \mid \bar{Y}_{nh}^h = y_i, \bar{V}_{nh}^h = v_k^n, \bar{X}_{nh}^h = x_j^n) \simeq u_{i,k,j}^n, \\ & \text{where } u_{i,k,j}^n = (u_{i,k,j}^n)_{i \in \mathcal{J}_M} \text{ solves the linear system} \\ & A_{k,j}^n u_{i,k,j}^n \\ & = \sum_{a,b \in \{u,d\}} p_{ab}(n, k, j) Bf \left(y + \frac{\rho_1}{\sigma_V} (v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2 \sqrt{v} (x_{j_b(n,j)}^{n+1} - x_j^n) \right) + d^n. \end{aligned}$$

Finally, if f is a function on the whole triple (y, v, x) , by using standard properties of the conditional expectation one gets

$$\begin{aligned} & \mathbb{E}(f(\bar{Y}_{(n+1)h}^h, \bar{V}_{(n+1)h}^h, \bar{X}_{(n+1)h}^h) \mid \bar{Y}_{nh}^h = y_i, \bar{V}_{nh}^h = v_k^n, \bar{X}_{nh}^h = x_j^n) \simeq u_{i,k,j}^n, \\ & \text{where } u_{i,k,j}^n = (u_{i,k,j}^n)_{i \in \mathcal{J}_M} \text{ solves the linear system} \\ & A_{k,j}^n u_{i,k,j}^n \\ & = \sum_{a,b \in \{u,d\}} p_{ab}(n, k, j) Bf \left(y + \frac{\rho_1}{\sigma_V} (v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2 \sqrt{v} (x_{j_b(n,j)}^{n+1} - x_j^n), v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1} \right) + d^n. \end{aligned} \quad (4.17)$$

4.2 Pricing European and American options

We are now ready to approximate the function P_h solution to the dynamic programming principle (4.1). We consider the discretization scheme $(\bar{Y}^h, \bar{V}^h, \bar{X}^h)$ discussed in Section 4.1 and we use the approximation (4.17) for the conditional expectations that have to be computed at each time step n . So, for every point $(y_i, v_k^n, x_j^n) \in \mathcal{Y}_M \times \mathcal{V}_n \times \mathcal{X}_n$, by (4.17) we have

$$\mathbb{E} \left(P_h((n+1)h, Y_{(n+1)h}^{nh, y_i, v_k^n, x_j^n}, V_{(n+1)h}^{nh, v_k^n}, X_{(n+1)h}^{nh, x_j^n}) \right) \simeq u_{i,k,j}^n$$

where $u_{i,k,j}^n = (u_{i,k,j}^n)_{i \in \mathcal{J}_M}$ solves the linear system

$$\begin{aligned} A_{k,j}^n u_{i,k,j}^n &= B \sum_{a,b \in \{u,d\}} p_{ab}(n, k, j) \times \\ & \times P_h \left((n+1)h, y + \frac{\rho_1}{\sigma_V} (v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2 \sqrt{v} (x_{j_b(n,j)}^{n+1} - x_j^n), v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1} \right) + d^n. \end{aligned} \quad (4.18)$$

We then define the approximated price $\tilde{P}_h(nh, y, v, x)$ for $(y, v, x) \in \mathcal{Y}_M \times \mathcal{V}_n \times \mathcal{X}_n$ and $n = 0, 1, \dots, N$ as

$$\begin{cases} \tilde{P}_h(T, y_i, v_{N,k}, x_{N,j}) = \Psi(y_i) & \text{and as } n = N-1, \dots, 0: \\ \tilde{P}_h(nh, y_i, v_k^n, x_j^n) = \max \left\{ \hat{\Psi}(y_i), e^{-(\sigma_r x_j^n + \varphi_{nh})h} \tilde{u}_{i,k,j}^n \right\} \end{cases} \quad (4.19)$$

in which $\tilde{u}_{i,k,j}^n = (\tilde{u}_{i,k,j}^n)_{i \in \mathcal{J}_M}$ is the solution to the system in (4.18) with P_h replaced by \tilde{P}_h .

Note that the system in (4.18) requires the knowledge of the function $y \mapsto \tilde{P}_h((n+1)h, y, v, x)$ in points y 's that do not necessarily belong to the grid \mathcal{Y}_M . Therefore, in practice we compute such a function by means of linear interpolations, working as follows. For fixed n, k, j, a, b , we set $I_{n,k,j,a,b}(i)$, $i \in \mathcal{J}_M$, as the index such that

$$y_i + \frac{\rho_1}{\sigma_V} (v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2 \sqrt{v_k^n} (x_{j_b(n,j)}^{n+1} - x_j^n) \in [y_{I_{n,k,j,a,b}(i)}, y_{I_{n,k,j,a,b}(i)+1}),$$

with $I_{n,k,j,a,b}(i) = -M$ if $y_i + \frac{\rho_1}{\sigma_V}(v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2\sqrt{v_k^n}(x_{j_b(n,j)}^{n+1} - x_j^n) < -M$ and $I_{n,k,j,a,b}(i) + 1 = M$ if $y_i + \frac{\rho_1}{\sigma_V}(v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2\sqrt{v_k^n}(x_{j_b(n,j)}^{n+1} - x_j^n) > M$. We set

$$q_{n,k,j,a,b}(i) = \frac{y_i + \frac{\rho_1}{\sigma_V}(v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2\sqrt{v_k^n}(x_{j_b(n,j)}^{n+1} - x_j^n) - y_{I_{n,k,j,a,b}(i)}}{\Delta y}.$$

Note that $q_{n,k,j,a,b}(i) \in [0, 1)$. We define

$$\begin{aligned} (\mathcal{I}_{a,b}\tilde{P}_h)((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}) &= \tilde{P}_h((n+1)h, y_{I_{n,k,j,a,b}(i)}, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}) (1 - q_{n,k,j,a,b}(i)) \\ &+ \tilde{P}_h((n+1)h, y_{I_{n,k,j,a,b}(i)+1}, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}) q_{n,k,j,a,b}(i) \end{aligned}$$

and we set

$$\begin{aligned} &\tilde{P}_h\left((n+1)h, y_i + \frac{\rho_1}{\sigma_V}(v_{k_a(n,k)}^{n+1} - v_k^n) + \rho_2\sqrt{v_k^n}(x_{j_b(n,j)}^{n+1} - x_j^n), v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}\right) \\ &= (\mathcal{I}_{a,b}\tilde{P}_h)((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}). \end{aligned}$$

Therefore, starting from (4.18), in practice the function $\tilde{u}_{i,k,j}^n = (\tilde{u}_{i,k,j}^n)_{i \in \mathcal{J}_M}$ in (4.19) is taken as the solution to the linear system

$$A_{k,j}^n \tilde{u}_{i,k,j}^n = B \sum_{a,b \in \{u,d\}} p_{ab}(n, k, j) (\mathcal{I}_{a,b}\tilde{P}_h)((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}) + d^n. \quad (4.20)$$

We can then state our final numerical procedure:

$$\begin{cases} \tilde{P}_h(T, y_i, v_{N,k}, x_{N,j}) = \Psi(y_i) & \text{and as } n = N-1, \dots, 0: \\ \tilde{P}_h(nh, y_i, v_k^n, x_j^n) = \max \left\{ \hat{\Psi}(y_i), e^{-(\sigma_r x_j^n + \varphi_{nh})h} \tilde{u}_{i,k,j}^n \right\} \end{cases} \quad (4.21)$$

$\tilde{u}_{i,k,j}^n = (\tilde{u}_{i,k,j}^n)_{i \in \mathcal{J}_M}$ being the solution to the system (4.20).

Remark 4.1 *In the case of infinite grid, that is $M = +\infty$, $i \mapsto I_{n,k,j,a,b}(i)$ is a translation: $I_{n,k,j,a,b}(i) = I_{n,k,j,a,b}(0) + i$. So, $y_i \mapsto (\mathcal{I}_{a,b}\tilde{P}_h)((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1})$ is just a linear convex combination of a translation of $y_i \mapsto \tilde{P}_h((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1})$.*

5 Stability analysis of the hybrid tree/finite-difference method

For the study of the stability, we consider a norm on the functions of (y, v, x) which is uniform with respect to the volatility and the interest rate components (v, x) and coincides with the standard l_2 norm with respect to the direction y (see next (5.6)). The choice of the l_2 norm allows one to perform a von Neumann analysis in the component y on the infinite grid $\mathcal{Y} = \{y_i = Y_0 + i\Delta y\}_{i \in \mathbb{Z}}$, that is, without truncating the domain and without imposing boundary conditions. Therefore, our stability analysis does not take into account boundary effects. This approach is extensively used in the literature, see e.g. [21], and yields good criteria on the robustness of the algorithm independently of the boundary conditions.

Let us first write down explicitly the scheme (4.21) on the infinite grid $\mathcal{Y} = \{y_i\}_{i \in \mathbb{Z}}$. For a fixed function $f = f(t, y, v, x)$, we set $g = f$ either $g = 0$ and we consider the numerical scheme given by

$$\begin{cases} F_h(T, y_i, v_{N,k}, x_{N,j}) = f(T, y_i, v_{N,k}, x_{N,j}) & \text{and as } n = N-1, \dots, 0: \\ F_h(nh, y_i, v_k^n, x_j^n) = \max \left\{ g(nh, y_i, v_k^n, x_j^n), e^{-(\sigma_r x_j^n + \varphi_{nh})h} u_{i,k,j}^n \right\} \end{cases} \quad (5.1)$$

where $u_{k,j}^n = (u_{i,k,j}^n)_{i \in \mathbb{Z}}$ is the solution to

$$\begin{aligned} & (\alpha_{n,k,j} - \beta_{n,k})u_{i-1,k,j}^n + (1 + 2\beta_{n,k})u_{i,k,j}^n - (\alpha_{n,k,j} + \beta_{n,k})u_{i+1,k,j}^n \\ &= \sum_{a,b \in \{d,u\}} p_{ab}(n,k,j) \times \left[(\mathfrak{I}_{a,b}F_h)((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}) + \right. \\ & \quad \left. + h\Delta y \sum_l \nu(\xi_l) ((\mathfrak{I}_{a,b}F_h)((n+1)h, y_{i+l}, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1}) - (\mathfrak{I}_{a,b}F_h)((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1})) \right], \end{aligned} \quad (5.2)$$

in which $\alpha_{n,k,j}$ and $\beta_{n,k,j}$ are the coefficients α and β defined in (4.13) when evaluated in the pair (v_k^n, x_j^n) . Note that (5.2) is simply the linear system (4.20) on the infinite grid, with $d^n \equiv 0$ (no boundary conditions are needed). Let us stress that in next Remark 5.2 we will see that, since $\beta_{n,k} \geq 0$, a solution to (5.2) does exist, at least for “nice” functions f . It is clear that the case $g = f$ is linked to the American algorithm whereas the case $g = 0$ is connected to the European one: (5.1) gives our numerical approximation of the function

$$F(t, y, v, x) = \begin{cases} \mathbb{E} \left(e^{-(\sigma_r \int_t^T X_s^{t,x} ds + \int_t^T \varphi_s ds)} f(T, Y_T^{t,y,v,x}, V_T^{t,v}, X_T^{t,x}) \right) & \text{if } g = 0, \\ \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E} \left(e^{-(\sigma_r \int_t^\tau X_s^{t,x} ds + \int_t^\tau \varphi_s ds)} f(\tau, Y_\tau^{t,y,v,x}, V_\tau^{t,v}, X_\tau^{t,x}) \right) & \text{if } g = f, \end{cases} \quad (5.3)$$

at times nh and in the points of the grid $\mathcal{Y} \times \mathcal{V}_n \times \mathcal{X}_n$.

5.1 The “discount truncated scheme” and its stability

In our stability analysis, we consider a numerical scheme which is a slightly modification of (5.1): we fix a (possibly large) threshold $L > 0$ and we consider the scheme

$$\begin{cases} F_h^L(T, y_i, v_{N,k}, x_{N,j}) = f(T, y_i, v_{N,k}, x_{N,j}) & \text{and as } n = N - 1, \dots, 0: \\ F_h^L(nh, y_i, v_k^n, x_j^n) = \max \left\{ g(nh, y_i, v_k^n, x_j^n), e^{-(\sigma_r x_j^n 1_{\{x_j^n > -L\}} + \varphi_{nh})h} u_{i,k,j}^n \right\} \end{cases} \quad (5.4)$$

with $g = f$ or $g = 0$, where $u_{k,j}^n = (u_{i,k,j}^n)_{i \in \mathbb{Z}}$ is the solution to (5.2), with $(\mathfrak{I}_{a,b}F_h)$ replaced by $(\mathfrak{I}_{a,b}F_h^L)$. Let us stress that the above scheme (5.4) really differs from (5.1) only when $\sigma_r > 0$ (stochastic interest rate). And in this case, in the discounting factor of (5.4) we do not allow x_j^n to run everywhere on its grid: in the original scheme (5.1), the exponential contains the term x_j^n whereas in the present scheme (5.4) we put $x_j^n 1_{\{x_j^n > -L\}}$, so we kill the points of the grid \mathcal{X}_n below the threshold $-L$. And in fact, (5.4) aims to numerically compute the function

$$F^L(t, y, v, x) = \begin{cases} \mathbb{E} \left(e^{-(\sigma_r \int_t^T X_s^{t,x} 1_{\{X_s^{t,x} > -L\}} ds + \int_t^T \varphi_s ds)} f(T, Y_T^{t,y,v,x}, V_T^{t,v}, X_T^{t,x}) \right) & \text{if } g = 0, \\ \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E} \left(e^{-(\sigma_r \int_t^\tau X_s^{t,x} 1_{\{X_s^{t,x} > -L\}} ds + \int_t^\tau \varphi_s ds)} f(\tau, Y_\tau^{t,y,v,x}, V_\tau^{t,v}, X_\tau^{t,x}) \right) & \text{if } g = f, \end{cases} \quad (5.5)$$

at times nh and in the points of the grid $\mathcal{Y} \times \mathcal{V}_n \times \mathcal{X}_n$. Recall that in practice h is small but fixed, therefore there is a natural threshold which actually comes on in practice (see for instance the tree given in Figure 1). And actually, in our numerical experiments we observe a real stability. However, we will discuss later on how much one can loose with respect to the solution of (5.1).

For $n = N, \dots, 0$, the scheme (5.4) gives back a function in the variables $(y, v, x) \in \mathcal{Y} \times \mathcal{V}_n \times \mathcal{X}_n$. Note that $\mathcal{V}_n \times \mathcal{X}_n \subset I_n^V \times I_n^X$, where

$$I_n^V = [v_0^n, v_n^n] \quad \text{and} \quad I_n^X = [x_0^n, x_n^n],$$

that is, the intervals between the smallest and the biggest node at time-step n :

$$v_0^n = \left(\sqrt{V_0} - \frac{\sigma_V}{2} n\sqrt{h} \right)^2 1_{\{\sqrt{V_0} - \frac{\sigma_V}{2} n\sqrt{h} > 0\}}, \quad v_n^n = \left(\sqrt{V_0} + \frac{\sigma_V}{2} n\sqrt{h} \right)^2, \quad x_0^n = -n\sqrt{h}, \quad x_n^n = n\sqrt{h}.$$

As n decreases to 0, the intervals I_n^V and I_n^X are becoming smaller and smaller and at time 0 they collapse to the single point $v_0^0 = V_0$ and $x_0^0 = X_0 = 0$ respectively. So, the norm we are going to define takes into account these facts: at time nh we consider for $\phi = \phi(t, y, v, x)$ the norm

$$\|\phi(nh, \cdot)\|_n = \sup_{(v,x) \in I_n^V \times I_n^X} \|\phi(nh, \cdot, v, x)\|_{l_2(\mathcal{Y})} = \sup_{(v,x) \in I_n^V \times I_n^X} \left(\sum_{i \in \mathbb{Z}} |\phi(nh, y_i, v, x)|^2 \Delta y \right)^{\frac{1}{2}}. \quad (5.6)$$

In particular,

$$\begin{aligned} \|\phi(0, \cdot)\|_0 &= \|\phi(0, \cdot, V_0, X_0)\|_{l_2(\mathcal{Y})} = \left(\sum_{i \in \mathbb{Z}} |\phi(y_i, V_0, X_0)|^2 \Delta y \right)^{1/2} \quad \text{and} \\ \|\phi(T, \cdot)\|_N &\leq \sup_{(v,x) \in \mathbb{R}_+ \times \mathbb{R}} \|\phi(y_i, v, x)\|_{l_2(\mathcal{Y})} = \sup_{(v,x) \in \mathbb{R}_+ \times \mathbb{R}} \left(\sum_{i \in \mathbb{Z}} |\phi(y_i, v, x)|^2 \Delta y \right)^{1/2}. \end{aligned}$$

We are now ready to give our stability result.

Theorem 5.1 *Let $f \geq 0$ and, in the case $g = f$, suppose that*

$$\sup_{t \in [0, T]} |f(t, y, v, x)| \leq \gamma_T |f(T, y, v, x)|,$$

for some $\gamma_T > 0$. Then, for every $L > 0$ the numerical scheme (5.4) is stable with respect to the norm (5.6):

$$\|F_h^L(0, \cdot)\|_0 \leq C_T^{N,L} \|F_h^L(T, \cdot)\|_N = C_T^{N,L} \|f(T, \cdot)\|_N, \quad \forall h, \Delta y,$$

where

$$C_T^{N,L} = \begin{cases} e^{2\lambda cT + \sigma_r LT - \sum_{n=1}^N \varphi_{nh} h} \xrightarrow{N \rightarrow \infty} C_T^L = e^{2\lambda cT + \sigma_r LT - \int_0^T \varphi_t dt} & \text{if } g = 0, \\ \max \left(\gamma_T, e^{2\lambda cT + \sigma_r LT - \sum_{n=1}^N \varphi_{nh} h} \right) \xrightarrow{N \rightarrow \infty} C_T^L = \max \left(\gamma_T, e^{2\lambda cT + \sigma_r LT - \int_0^T \varphi_t dt} \right) & \text{if } g = f, \end{cases}$$

in which $c > 0$ is such that $\sum_l \nu(\xi_l) \Delta y \leq \lambda c$.

Proof. In order to weaken the notation, we set $g_{i,k,j}^n = g(nh, y_i, v_k^n, x_j^n)$ and, similarly, $F_{i,k,j}^n = F_h^L(nh, y_i, v_k^n, x_j^n)$, $(\mathfrak{I}_{a,b} F_h^{n+1})_{i,k_a,j_b} = (\mathfrak{I}_{a,b} F_h^L)((n+1)h, y_i, v_{k_a(n,k)}^{n+1}, x_{j_b(n,j)}^{n+1})$ (we have also dropped the dependence on L). The scheme (5.4) says that, at each time step $n < N$ and for each fixed $0 \leq k, j \leq n$,

$$F_{i,k,j}^n = \max \left\{ g_{i,k,j}^n, e^{-(\sigma_r x_j^n 1_{\{x_j^n > -L\}} + \varphi_{nh})h} u_{i,k,j}^n \right\}, \quad (5.7)$$

where, according to (5.2), $u_{i,k,j}^n$ solves

$$\begin{aligned} &(\alpha_{n,k,j} - \beta_{n,k}) u_{i-1,k,j}^n + (1 + 2\beta_{n,k}) u_{i,k,j}^n - (\alpha_{n,k,j} + \beta_{n,k}) u_{i+1,k,j}^n \\ &= \sum_{a,b \in \{d,u\}} p_{ab}(n, k, j) \left((\mathfrak{I}_{a,b} F_h^{n+1})_{i,k_a,j_b} + h \Delta y \sum_l \nu(\xi_l) [(\mathfrak{I}_{a,b} F_h^{n+1})_{i+l,k_a,j_b} - (\mathfrak{I}_{a,b} F_h^{n+1})_{i,k_a,j_b}] \right). \end{aligned} \quad (5.8)$$

Let $\mathfrak{F}\varphi$ denote the Fourier transform of $\varphi \in l_2(\mathcal{Y})$, that is,

$$\mathfrak{F}\varphi(\theta) = \frac{\Delta y}{\sqrt{2\pi}} \sum_{s \in \mathbb{Z}} \varphi_s e^{-is\Delta y \theta}, \quad \theta \in \mathbb{R},$$

\mathbf{i} denoting the imaginary unit. We get from (5.8)

$$\begin{aligned} & \left((\alpha_{n,k,j} - \beta_{n,k}) e^{-\mathbf{i}\theta\Delta y} + 1 + 2\beta_{n,k} - (\alpha_{n,k,j} + \beta_{n,k}) e^{\mathbf{i}\theta\Delta y} \right) \mathfrak{F}u_{k,j}^n(\theta) \\ &= \left(1 + h\Delta y \sum_l \nu(\xi_l) (e^{\mathbf{i}l\theta\Delta y} - 1) \right) \sum_{a,b \in \{d,u\}} p_{ab}(n, k, j) \mathfrak{F}(\mathfrak{I}_{a,b} F^{n+1})_{k_a, j_b}(\theta). \end{aligned} \quad (5.9)$$

Note that

$$\begin{aligned} & |(\alpha_{n,k,j} - \beta_{n,k}) e^{-\mathbf{i}\theta\Delta y} + 1 + 2\beta_{n,k} - (\alpha_{n,k,j} + \beta_{n,k}) e^{\mathbf{i}\theta\Delta y}| \\ & \geq |\Re[(\alpha_{n,k,j} - \beta_{n,k}) e^{-\mathbf{i}\theta\Delta y} + 1 + 2\beta_{n,k} - (\alpha_{n,k,j} + \beta_{n,k}) e^{\mathbf{i}\theta\Delta y}]| \\ & = 1 + 2\beta_{n,k}(1 - \cos(\theta\Delta y)) \geq 1, \end{aligned}$$

for every $\theta \in [0, 2\pi)$ (recall that $\beta_{n,k} \geq 0$). And since $\sum_l \nu(\xi_l) \Delta y \leq \lambda c$, we obtain

$$\begin{aligned} |\mathfrak{F}u_{k,j}^n(\theta)| & \leq \left(1 + h\Delta y \sum_{l \in \mathbb{Z}} |e^{\mathbf{i}l\theta\Delta y} - 1| \nu(\xi_l) \right) \sum_{a,b \in \{d,u\}} p_{ab}(n, k, j) |\mathfrak{F}(\mathfrak{I}_{a,b} F^{n+1})_{k_a, j_b}(\theta)| \\ & \leq (1 + 2\lambda ch) \sum_{a,b \in \{d,u\}} p_{ab}(n, k, j) |\mathfrak{F}(\mathfrak{I}_{a,b} F^{n+1})_{k_a, j_b}(\theta)|. \end{aligned}$$

Therefore,

$$\|\mathfrak{F}u_{k,j}^n\|_{L^2([0, 2\pi), \text{Leb})} \leq (1 + 2\lambda ch) \sum_{a,b \in \{d,u\}} p_{ab}(n, k, j) \|\mathfrak{F}(\mathfrak{I}_{a,b} F^{n+1})_{k_a, j_b}\|_{L^2([0, 2\pi), \text{Leb})}.$$

We use now the Parseval identity $\|\mathfrak{F}\varphi\|_{L^2([0, 2\pi), \text{Leb})} = \|\varphi\|_{l_2(\mathcal{Y})}$ and we get

$$\begin{aligned} \|u_{\cdot, k, j}^n\|_{l_2(\mathcal{Y})} & \leq (1 + 2\lambda ch) \sum_{a,b \in \{d,u\}} p_{ab}(n, k, j) \|(\mathfrak{I}_{a,b} F^{n+1})_{\cdot, k_a, j_b}\|_{l_2(\mathcal{Y})} \\ & = (1 + 2\lambda ch) \sum_{a,b \in \{d,u\}} p_{ab}(n, k, j) \|F_{\cdot, k_a, j_b}^{n+1}\|_{l_2(\mathcal{Y})}, \end{aligned}$$

the first equality following from the fact that $i \mapsto (\mathfrak{I}_{a,b} F^{n+1})_{i, k_a, j_b}$ is a linear convex combination of translations of $i \mapsto F_{i, k_a, j_b}^{n+1}$ (see Remark 4.1). This gives

$$\sup_{0 \leq k, j \leq n} \|e^{-(\sigma_r x_j^n 1_{\{x_j^n > -L\}} + \varphi_{nh})h} u_{\cdot, k, j}^n\|_{l_2(\mathcal{Y})} \leq (1 + 2\lambda ch) e^{\sigma_r Lh - \varphi_{nh}h} \sup_{0 \leq k, j \leq n+1} \|F_{\cdot, k, j}^{n+1}\|_{l_2(\mathcal{Y})}$$

and from (5.7), we obtain

$$\sup_{0 \leq k, j \leq n} \|F_{\cdot, k, j}^n\|_{l_2(\mathcal{Y})} \leq \max \left(\sup_{0 \leq k, j \leq n} \|g_{\cdot, k, j}^n\|_{l_2(\mathcal{Y})}, (1 + 2\lambda ch) e^{\sigma_r Lh - \varphi_{nh}h} \sup_{0 \leq k, j \leq n+1} \|F_{\cdot, k, j}^{n+1}\|_{l_2(\mathcal{Y})} \right).$$

We now continue assuming that $g = f$, the case $g = 0$ following in a similar way. So,

$$\sup_{0 \leq k, j \leq n} \|F_{\cdot, k, j}^n\|_{l_2(\mathcal{Y})} \leq \max \left(\gamma_T \|f(T, \cdot)\|_N, (1 + 2\lambda ch) e^{\sigma_r Lh - \varphi_{nh}h} \sup_{0 \leq k, j \leq n+1} \|F_{\cdot, k, j}^{n+1}\|_{l_2(\mathcal{Y})} \right).$$

For $n = N - 1$ we then obtain

$$\sup_{0 \leq k, j \leq n} \|F_{\cdot, k, j}^{N-1}\|_{l_2(\mathcal{Y})} \leq \max \left(\gamma_T \|f(T, \cdot)\|_N, (1 + 2\lambda ch) e^{\sigma_r Lh - \varphi_{(N-1)h} h} \|f(T, \cdot)\|_N \right)$$

and by iterating the above inequalities, we finally get

$$\|F^0\|_0 = \|F_{\cdot, 0, 0}^0\|_{l_2(\mathcal{Y})} \leq \max \left(\gamma_T \|f(T, \cdot)\|_N, (1 + 2\lambda ch)^N e^{N\sigma_r Lh - \sum_{n=1}^N \varphi_{nh} h} \|f(T, \cdot)\|_N \right).$$

□

Remark 5.2 *In the above proof we have actually proved that, as n varies, the solution $u_{\cdot, k, j}^n$ to the infinite linear system (5.2) does exist and is unique if $\|f(T, \cdot)\|_N < \infty$. In fact, starting from equality (5.9), we define the function $\psi_{k, j}(\theta)$, $\theta \in [0, 2\pi)$, by*

$$\begin{aligned} & \left((\alpha_{n, k, j} - \beta_{n, k}) e^{-i\theta \Delta y} + 1 + 2\beta_{n, k} - (\alpha_{n, k, j} + \beta_{n, k}) e^{i\theta \Delta y} \right) \psi_{k, j}(\theta) \\ &= \left(1 + h\Delta y \sum_l \nu(\xi_l) (e^{il\theta \Delta y} - 1) \right) \sum_{a, b \in \{d, u\}} p_{ab}(n, k, j) \mathfrak{F}(\mathfrak{I}_{a, b} F^{n+1})_{k_a, j_b}(\theta). \end{aligned}$$

As noticed in the proof of Proposition 5.1, the factor multiplying $\psi_{k, j}(\theta)$ is different from zero because $\beta_{n, k} \geq 0$. So, the definition of $\psi_{k, j}$ is well posed and moreover, $\psi_{k, j} \in L^2([0, 2\pi), \text{Leb})$. We now set $u_{\cdot, k, j}^n$ as the inverse Fourier transform of $\psi_{k, j}$, that is,

$$u_{l, k, j}^n = \frac{1}{\Delta y \sqrt{2\pi}} \int_0^{2\pi} \psi_{k, j}(\theta) e^{il\theta \Delta y} d\theta, \quad l \in \mathbb{Z}.$$

Straightforward computations give that $u_{\cdot, k, j}^n$ fulfils the equation system (5.2).

Of course, Theorem 5.1 gives a stability property for the scheme introduced in [9] for the Heston-Hull-White model (just take $\lambda = 0$ - no jumps are considered). Moreover, for the standard Bates model, that is, $\sigma_r = 0$ (deterministic interest rate), Theorem 5.1 applies to the original (untruncated) scheme (5.1).

5.2 Back to the original scheme (5.1)

Let us now discuss what may happen when one introduces the threshold L . We recall that the original scheme (5.1) gives the numerical approximation of the function F in (5.3) whereas the discount truncated scheme (5.4) aims to numerically compute the function F^L in (5.5). Proposition 5.3 below shows that, under standard hypotheses, F^L tends to F as $L \rightarrow \infty$ very fast. This means that, in practice, we loose very few in using (5.4) in place of (5.1).

Proposition 5.3 *Suppose that $f = f(t, y, v, x)$ has a polynomial growth in the variables (y, v, x) , uniformly in $t \in [0, T]$. Let F and F^L , with $L > 0$, be defined in (5.3) and (5.5) respectively. Then there exist positive constants c_T and $C_T(y, v, x)$ such that for every $L > 0$*

$$|F(t, y, v, x) - F^L(t, y, v, x)| \leq \sigma_r C_T(y, v, x) e^{-c_T |L + x e^{-\kappa_r (T-t)}|^2},$$

for every $t \in [0, T]$ and $(y, v, x) \in \mathbb{R} \times \mathbb{R}_+ \times \mathbb{R}$.

Proof. In the following, C denotes a positive constant, possibly changing from line to line. We have

$$|F(t, y, v, x) - F^L(t, y, v, x)| \leq \sigma_r C \mathbb{E} \left(\sup_{t \leq u \leq T} |f(u, Y_u^{t,y,v,x}, V_u^{t,v}, X_u^{t,v})| \times e^{-\sigma_r \int_t^u X_s^{t,x} 1_{\{X_s^{t,x} > -L\}} ds} \times \left| e^{-\sigma_r \int_t^u X_s^{t,x} 1_{\{X_s^{t,x} < -L\}} ds} - 1 \right| \right).$$

Set

$$\tau_{-L}^{t,x} = \inf\{s \geq t : X_s^{t,x} \leq -L\}.$$

One has $1_{\{X_s^{t,x} < -L\}} \leq 1_{\{\tau_{-L}^{t,x} < s\}}$ so that, for $u \leq T$,

$$0 \leq -\sigma_r \int_t^u X_s^{t,x} 1_{\{X_s^{t,x} < -L\}} ds \leq \sigma_r \int_t^u |X_s^{t,x}| 1_{\{\tau_{-L}^{t,x} < s\}} ds \leq \sigma_r \int_t^T |X_s^{t,x}| ds 1_{\{\tau_{-L}^{t,x} \leq T\}}.$$

Then,

$$\begin{aligned} & \sup_{t \leq u \leq T} e^{-\sigma_r \int_t^u X_s^{t,x} 1_{\{X_s^{t,x} > -L\}} ds} \left| e^{-\sigma_r \int_t^u X_s^{t,x} 1_{\{X_s^{t,x} < -L\}} ds} - 1 \right| \\ & \leq e^{\sigma_r \int_t^T |X_s^{t,x}| ds} \left(e^{\sigma_r \int_t^T |X_s^{t,x}| ds} - 1 \right) 1_{\{\tau_{-L}^{t,x} \leq T\}} \leq 2e^{2\sigma_r \int_t^T |X_s^{t,x}| ds} 1_{\{\tau_{-L}^{t,x} \leq T\}} \end{aligned}$$

By inserting,

$$\begin{aligned} & |F(t, y, v, x) - F^L(t, y, v, x)| \\ & \leq \sigma_r C \mathbb{E} \left(\sup_{t \leq u \leq T} |f(u, Y_u^{t,y,v,x}, V_u^{t,v}, X_u^{t,v})| e^{2\sigma_r \int_t^T |X_s^{t,x}| ds} 1_{\{\tau_{-L}^{t,x} \leq T\}} \right) \\ & \leq \sigma_r C \mathbb{E} \left(\sup_{t \leq u \leq T} |f(u, Y_u^{t,y,v,x}, V_u^{t,v}, X_u^{t,v})|^2 e^{4\sigma_r \int_t^T |X_s^{t,x}| ds} \right)^{1/2} \mathbb{P}(\tau_{-L}^{t,x} \leq T)^{1/2}. \end{aligned}$$

Since f has a polynomial growth in the space variables, uniformly in the time variable, by standard estimates one gets that $\sup_{t \leq u \leq T} |f(u, Y_u^{t,y,v,x}, V_u^{t,v}, X_u^{t,v})|$ has all moments. Moreover, for a Brownian motion W , $\sup_{0 \leq s \leq T} |W_s|$ has finite exponential moments of any order, and this gives that $e^{4\sigma_r \int_t^T |X_s^{t,x}| ds}$ has finite moments of any order. It follows that

$$|F(t, y, v, x) - F^L(t, y, v, x)| \leq C \mathbb{P}(\tau_{-L}^{t,x} \leq T)^{1/2}.$$

As for the above probability, recall that $X_s^{t,x} = xe^{-\kappa_r(s-t)} + \int_t^s e^{-\kappa_r(s-u)} dW_u^2$ so that

$$\begin{aligned} \mathbb{P}(\tau_{-L}^{t,x} \leq T) &= \mathbb{P} \left(\inf_{s \in [t, T]} X_s^{t,x} < -L \right) = \mathbb{P} \left(\inf_{s \in [t, T]} \left(xe^{-\kappa_r(s-t)} + \int_t^s e^{-\kappa_r(s-u)} dW_u^2 \right) < -L \right) \\ &\leq \mathbb{P} \left(\sup_{s \in [t, T]} \left| \int_t^s e^{\kappa_r u} dW_u^2 \right| > L + xe^{-\kappa_r(T-t)} \right) \leq 2 \exp \left(- \frac{|L + xe^{-\kappa_r(T-t)}|^2}{2 \int_t^T e^{2\kappa_r u} du} \right). \end{aligned}$$

By inserting above, we get the result. \square

5.3 Further remarks

As already stressed, the introduction of the threshold $-L$ allows one to handle the discount term. In order to let the discount disappear, an approach consists in the use of a transformed function, as developed by several authors (see e.g. Haentjens and in't Hout [25] and references therein). This is a nice fact for European options (PIDE problem), being on the contrary a non definitive tool when dealing with American options (obstacle PIDE problem). Let us see why.

First of all, let us come back to the model for the triple (Y, V, X) , see (2.5). The infinitesimal generator is

$$\begin{aligned} L_t u = & \left(\sigma_r x + \varphi_t - \eta - \frac{1}{2}v \right) \partial_y u + \kappa_V (\theta_V - v) \partial_v u - \kappa_r x \partial_x u \\ & + \frac{1}{2} \left(v \partial_{yy}^2 u + \sigma_V^2 v \partial_{vv}^2 u + \partial_{xx}^2 u + 2\rho_1 \sigma_V v \partial_{yv}^2 u + 2\rho_2 \sqrt{v} \partial_{yx}^2 u \right) \\ & + \int_{-\infty}^{+\infty} [u(t, y + \xi; v, x) - u(t, y; v, x)] \nu(\xi) d\xi. \end{aligned} \quad (5.10)$$

We set

$$G(t, x) = \mathbb{E} \left(e^{-\sigma_r \int_t^T X_s^{t,x} ds} \right).$$

Recall that (see e.g. [31])

$$G(t, x) = e^{-x\sigma_r \Lambda(t, T) - \frac{\sigma_r^2}{2\kappa_r} (\Lambda(t, T) - T + t) - \frac{\sigma_r^2}{4\kappa_r} \Lambda^2(t, T)}, \quad \Lambda(t, T) = \frac{1 - e^{-\kappa_r(T-t)}}{\kappa_r} \quad (5.11)$$

and, moreover, G solves the PDE

$$\begin{aligned} \partial_t G - \kappa_r x \partial_x G + \frac{1}{2} \partial_{xx}^2 G - \sigma_r x G &= 0, \quad t \in [0, T], x \in \mathbb{R}, \\ G(T, x) &= 1. \end{aligned} \quad (5.12)$$

Lemma 5.4 *Let L_t denote the infinitesimal generator in (5.10). Set $\bar{u} = u \cdot G^{-1}$. Then*

$$\partial_t u + L_t u - xu = G(\partial_t \bar{u} + \bar{L}_t \bar{u}),$$

where

$$\bar{L}_t = L_t - \sigma_r \frac{1 - e^{-\kappa_r(T-t)}}{\kappa_r} [\rho_2 \sqrt{v} \partial_y + \partial_x].$$

Proof. Since G depends on t and x only, straightforward computations give

$$\partial_t u + L_t u - xu = G[\partial_t \bar{u} + L_t \bar{u}] + \partial_x G(t, x) [\rho_2 \sqrt{v} \partial_y \bar{u} + \partial_x \bar{u}] + \bar{u} [\partial_t G - \kappa_r x \partial_x G + \frac{1}{2} \partial_{xx}^2 G - \sigma_r x G].$$

By (5.12), the last term is null. The statement now follows by observing that $\partial_x \ln G(t, x) = -\sigma_r \frac{1 - e^{-\kappa_r(T-t)}}{\kappa_r}$. \square

We notice that the operator \bar{L}_t in Lemma 5.4 is the infinitesimal generator of the jump-diffusion process $(\bar{Y}, \bar{V}, \bar{X})$ which solves the stochastic differential equation as in (2.5), with the same diffusion coefficients and jump-terms but with the new drift coefficients

$$\mu_{\bar{Y}}(t, v, x) = \mu_Y(v, x) - \sigma_r \frac{1 - e^{-\kappa_r(T-t)}}{\kappa_r} \rho_2 \sqrt{v}, \quad \mu_{\bar{V}}(v) \equiv \mu_V(v), \quad \mu_{\bar{X}}(x) = \mu_X(t, x) - \sigma_r \frac{1 - e^{-\kappa_r(T-t)}}{\kappa_r}.$$

Let us first discuss the scheme (5.1) for F in the case $g = 0$ (European options). By passing to the associated PIDE, Lemma 5.4 says that

$$F(t, y, v, x) = G(t, x) \bar{F}(t, y, v, x),$$

where

$$\bar{F}(t, y, v, x) = \mathbb{E}(e^{-\int_t^T \varphi_s ds} f(T, \bar{Y}_T^{t,y,v,x}, \bar{V}_T^{t,v}, \bar{X}_T^{t,x})).$$

Therefore, in practice one has to numerically evaluate the function \bar{F} . By using our hybrid tree/finite-difference approach, this means to consider the scheme in (5.4), with the new coefficient $\bar{\alpha}_{n,k,j}$ (written starting from the new drift coefficients) but with a discount depending on the (deterministic) function φ only, that is, with $e^{-(\sigma_r x_j^n 1_{\{x_j^n > -L\}} + \varphi_{nh})h}$ replaced by $e^{-\varphi_{nh}h}$. And the proof of the Proposition 5.1 shows that one gets

$$\|\bar{F}_h(0, \cdot)\|_0 \leq \max(\gamma_T, e^{2\lambda cT - \sum_{n=0}^N \varphi_{nh}h}) \|f(T, \cdot)\|_N.$$

In other words, by using a suitable transformation, the European scheme is always stable and no thresholds are needed.

Let us discuss now the American case, that is, the scheme (5.1) with $g = f$. One could think to use the above transformation in order to get rid of the exponential depending on the process X . Set again

$$\bar{F}(t, y, v, x) = G(t, x)^{-1} F(t, y, v, x).$$

By using the associated obstacle PIDE problem, Lemma 5.4 suggests that

$$\bar{F}(t, y, v, x) = \sup_{\tau \in \mathcal{T}_{t,T}} \mathbb{E}(e^{-\int_t^\tau \varphi_s ds} \bar{f}(\tau, \bar{Y}_\tau^{t,y,v,x}, \bar{V}_\tau^{t,v}, \bar{X}_\tau^{t,x})), \quad \text{with} \quad \bar{f}(t, y, v, x) = G^{-1}(t, x) f(t, y, v, x).$$

So, in order to numerically compute \bar{F} , one needs to set up the scheme (5.4) with the new coefficient $\bar{\alpha}_{n,k,j}$, with f replaced by \bar{f} , $g = \bar{f}$ and with the discounting factor $e^{-(\sigma_r x_j^n 1_{\{x_j^n > -L\}} + \varphi_{nh})h}$ replaced by $e^{-\varphi_{nh}h}$. So, again one is able to cancel the unbounded part of the discount. Nevertheless, the unpleasant point is that even if $\|f(T, \cdot)\|_N$ has a bound which is uniform in N then $\|\bar{f}(T, \cdot)\|_N$ may have not because $G^{-1}(t, x)$ has an exponential containing x , see (5.11). In other words, the unboundedness problem appears now in the obstacle.

6 The hybrid Monte Carlo and tree/finite-difference approach algorithms in practice

The present section is devoted to our numerical experiments. We first resume the main steps of our algorithms and then we present several numerical tests.

6.1 A schematic sketch of the main computational steps in our algorithms

To summarize, we resume here the main computational steps of the two proposed algorithms.

First, the procedures need the following preprocessing steps, concerning the construction of the bivariate tree:

- (T1) define a discretization of $[0, T]$ in N subintervals $[nh, (n+1)h]$, $n = 0, \dots, N-1$, with $h = T/N$;
- (T2) for the process V , set the binomial tree v_k^n , $0 \leq k \leq n \leq N$, by using (3.1), then compute the jump nodes $k_a(n, k)$ and the jump probabilities $p_a^V(n, k)$, $a \in \{u, d\}$, by using (3.3)-(3.4) and (3.7);

- (T3) for the process X , set the binomial tree x_j^n , $0 \leq j \leq N$, by using (3.2), then compute the jump nodes $j_b(n, j)$ and the jump probabilities $p_b^X(n, j)$, $b \in \{u, d\}$, by using (3.5)-(3.6) and (3.8);
- (T4) for the 2-dimensional process (V, X) , merge the binomial trees in the bivariate tree (v_k^n, x_j^n) , $0 \leq k, j \leq n \leq N$, by using (3.9), then compute the jump-nodes $(k_a(n, k), j_b(n, j))$ and the transitions probabilities $p_{ab}(n, k, j)$, $(a, b) \in \{d, u\}$, by using (3.10).

The bivariate tree for (V, X) is now settled. Our hybrid tree/finite-difference algorithm can be resumed as follows:

- (FD1) set a mesh grid y_i for the solution of all the PIDEs;
- (FD2) for each node $(v_{N,k}, x_{N,j})$, $0 \leq k, j \leq N$, compute the option prices at maturity for each y_i , $i \in \mathcal{Y}_M$, by using the payoff function;
- (FD3) for $n = N - 1, \dots, 0$: for each (v_k^n, x_j^n) , $0 \leq k, j \leq n$, compute the option prices for each $y_i \in \mathcal{Y}_M$, by solving the linear system (4.20).

Notice that, at each time step n , we need only the one-step PIDE solution in the time interval $[nh, (n+1)h]$. Moreover, both the (constant) PIDE coefficients and the Cauchy final condition change according to the position of the volatility and the interest rate components on the bivariate tree at time step n .

We conclude by briefly recalling the main steps of the hybrid Monte Carlo method:

- (MC1) let the chain $(\hat{V}_n^h, \hat{X}_n^h)$ evolve for $n = 1, \dots, N$, following the probability structure in (T4);
- (MC2) generate $\Delta_1, \dots, \Delta_N$ i.i.d. standard normal r.v.'s independent of the noise driving the chain (\hat{V}^h, \hat{X}^h) ;
- (MC3) generate K_h^1, \dots, K_h^N i.i.d. positive Poisson r.v.'s of parameter λh , independent of both the chain (\hat{V}^h, \hat{X}^h) and the Gaussian r.v.'s $\Delta_1, \dots, \Delta_N$, and for every $n = 1, \dots, N$, if $K_h^n > 0$ simulate the corresponding amplitudes $\log(1 + J_1^n), \dots, \log(1 + J_{K_h^n}^n)$;
- (MC4) starting from $\hat{Y}_0^h = Y_0$, compute the approximate values \hat{Y}_n^h , $1 \leq n \leq N$, by using (3.14);
- (MC5) following the desired Monte Carlo method (European or Longstaff-Schwartz algorithm [33] in the case of American options), repeat the above simulation scheme and compute the option price.

Remark 6.1 *In next Section 6.2 we develop numerical experiments also in the standard Bates model, that is, with a constant interest rate. Recall that in the standard Bates model the dynamic reduces to*

$$\begin{aligned} \frac{dS_t}{S_{t-}} &= (r - \eta)dt + \sqrt{V_t} dZ_t^S + dH_t, \\ dV_t &= \kappa_V(\theta_V - V_t)dt + \sigma_V \sqrt{V_t} dZ_t^V, \end{aligned} \tag{6.1}$$

with $S_0, V_0 > 0$, $r \geq 0$ constant parameters, $d\langle Z^S, Z^V \rangle_t = \rho dt$, $|\rho| < 1$ and H_t is the compound Poisson process already introduced in Section 2, see (2.2). We can apply our hybrid approach to this case as well: it suffices just to follow the computational steps listed above except for the construction of the binomial tree for the process X . Consequently, we do not need the bivariate tree for (V, X) , specifically we omit steps (T3)-(T4) and we replace step (MC1) with

(MC1') let the chain \hat{V}_n^h evolve for $n = 1, \dots, N$, following the probability structure in (T2).

And of course, in all computations we set equal to 0 the parameters involved in the dynamics for r , except for the starting value r_0 . In particular, we have $\sigma_r = 0$ and $\varphi_t = r_0$ for every t .

Remark 6.2 We observe that in order to compute the option price by the hybrid tree/finite-difference procedure, in step (FD3) we need to solve many times the tridiagonal system (4.20). This is typically solved by the LU-decomposition method in $O(M)$ operations (recall that the total number of the grid values $y_i \in \mathcal{Y}_M$ is $2M + 1$). However, due to the approximation of the integral term (4.9), at each time step $n < N$ we have to compute the sum

$$\sum \tilde{u}_{i+l}^{n+1} \nu(\xi_l), \quad (6.2)$$

which is the most computationally expensive step of this part of the algorithm: when applied directly, it requires $O(M^2)$ operations. Following the Premia software implementation [37], in our numerical tests we use the Fast Fourier Transform to compute the term (6.2) and the computational costs of this step reduce to $O(M \log M)$. According to the Bates (respectively the Bates-Hull-White) model, the hybrid algorithm requires $N(N+1)/2$ (respectively $(N(N+1)/2)^2$) resolutions of linear systems, each of them having a linear complexity. Therefore, the overall complexity is $O(N^2 M \log M)$ (respectively $O(N^4 M \log M)$).

6.2 Numerical results

We develop several numerical results in order to assess the efficiency and the robustness of the hybrid tree/finite-difference method and the hybrid Monte Carlo method in the case of plain vanilla options. The Monte Carlo results derive from our hybrid simulations and, for American options, the use of the Monte Carlo algorithm by Longstaff and Schwartz in [33].

We first provide results for the standard Bates model (see Remark 6.1) and secondly, for the case in which the interest rate process is assumed to be stochastic, see (2.1).

Following Chiarella *et al.* [16], in our numerical tests we assume that the jumps for the log-returns are normal, that is,

$$\log(1 + J_1) \sim N\left(\gamma - \frac{1}{2}\delta^2, \delta^2\right), \quad (6.3)$$

N denoting the Gaussian law (we also notice that the results in [16] correspond to the choice $\gamma = 0$). In Section 6.2.1, we first compare our results with the ones provided in Chiarella *et al.* [16]. Then in Section 6.2.2 we study options with large maturities and when the Feller condition is not fulfilled. Finally, Section 6.2.3 is devoted to test experiments for European and American options in the Bates model with stochastic interest rate. The codes have been written by using the C++ language and the computations have been all performed in double precision on a PC 2,9 GHz Intel Core I5 with 8 Gb of RAM.

6.2.1 The standard Bates model

We refer here to the standard Bates model as in (6.1). In the European and American option contracts we are dealing with, we consider the following set of parameters, already used in the numerical results provided in Chiarella *et al.* [16]: initial price $S_0 = 80, 90, 100, 110, 120$, strike price $K = 100$, maturity $T = 0.5$; (constant) interest rate $r = 0.03$, dividend rate $\eta = 0.05$; initial volatility $V_0 = 0.04$, long-mean $\theta_V = 0.04$, speed of mean-reversion $\kappa_V = 2$, vol-vol $\sigma_V = 0.4$, correlation $\rho = -0.5, 0.5$; intensity $\lambda = 5$, jump parameters $\gamma = 0$ and $\delta = 0.1$ (recall (6.3)). It is known that the case $\rho > 0$ may lead to

moment explosion, see. e.g. [4]. Nevertheless, we report here results for this case as well, for the sake of comparisons with the study in Chiarella *et al.* [16].

In order to numerically solve the PIDE using the finite difference scheme, we first localize the variables and the integral term to bounded domains. We use for this purpose the estimates for the localization domain and the truncation of large jumps given by Voltchkova and Tankov [42]. For example, for the previous model parameters the PIDE problem is solved in the finite interval $[\ln S_0 - 1.59, \ln S_0 + 1.93]$.

The numerical study of the hybrid tree/finite-difference method **HTFD** is split in two cases

HTFDa: time steps $N_t = 50$ and varying mesh grid $\Delta y = 0.01, 0.005, 0.0025, 0.00125$;

HTFDb: time steps $N_t = 100$ and varying mesh grid $\Delta y = 0.01, 0.005, 0.0025, 0.00125$.

Concerning the Monte Carlo method, we give the results from the hybrid simulation scheme in Section 3.3, that we call **HMC**. We give comparisons with the accurate third-order Alfonsi [2] discretization scheme for the CIR stochastic volatility process and by using an exact scheme for the interest rate. In addition, we simulate the jump component in the standard way. The resulting Monte Carlo scheme is here called **AMC**. We consider varying number of Monte Carlo iterations N_{MC} and two cases for the number of time discretization steps iterations:

HMCa and **AMCa**: $N_t = 50$ and $N_{MC} = 10000, 50000, 100000, 200000$;

HMCb and **AMCb**: $N_t = 100$ and $N_{MC} = 10000, 50000, 100000, 200000$.

All Monte Carlo results include the associated 95% confidence interval.

Table 1 reports European call option prices. Comparisons are given with a benchmark value obtained using the Carr-Madan pricing formula **CF** in [15] that applies Fast Fourier Transform methods (see the Premia software implementation [37]).

In Table 2 we provide results for American call option prices. In this case we compare with the values obtained by using the method of lines in [17], called **MOL**, with mesh parameters 200 time-steps, 250 volatility lines, 2995 asset grid points, and the **PSOR** method with mesh parameters 1000, 3000, 6000 that Chiarella *et al.* [16] used as the true solution. Moreover, we consider the Longstaff-Schwartz [33] Monte Carlo algorithm both for **AMC** and **HMC**. In particular

HMCLSa and **AMCLSa**: 10 exercise dates, $N_t = 50$ and $N_{MC} = 10000, 50000, 100000, 200000$;

HMCLsb and **AMCLsb**: 20 exercise dates, $N_t = 100$ and $N_{MC} = 10000, 50000, 100000, 200000$.

Tables 3 and 4 refer to the computational time cost (in seconds) of the various algorithms for $\rho = -0.5$ in the European and American case respectively.

In order to study the convergence behavior of our approach **HTFD**, we consider the convergence ratio proposed in [20], defined as

$$\text{ratio} = \frac{P_N - P_{\frac{N}{4}}}{P_N - P_{\frac{N}{2}}}, \quad (6.4)$$

where P_N denotes here the approximated price obtained with $N = N_t$ number of time steps. Recall that $P_N = O(N^{-\alpha})$ means that $\text{ratio} = 2^\alpha$. Table 5 suggests that the convergence ratio for **HTDFb** is approximatively linear.

The numerical results in Table 1-4 show that **HTFD** is accurate, reliable and efficient for pricing European and American options in the Bates model. Moreover, our hybrid Monte Carlo algorithm **HMC** appears to be competitive with **AMC**, that is the one from the simulations by Alfonsi [2]: the

numerical results are similar in term of precision and variance but **HMC** is definitely better from the computational times point of view. Additionally, because of its simplicity, **HMC** represents a real and interesting alternative to **AMC**.

As a further evidence of the accuracy of our hybrid methods, in Figure 2 and 3 we study the shapes of implied volatility smiles across moneyness $\frac{K}{S_0}$ and maturities T using **HTFDa** and **HMCa**. We compare the graphs with the results from the benchmark values **CF**. The parameters used for these two tests cases are: $T = 0.5$, $S_0 = 100$, $K = 100$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta_V = 0.04$, $\kappa_V = 2$, $\sigma_V = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5$.

(a)

$\rho = -0.5$	Δy	HTFDa	HTFDb	CF	N_{MC}	HMCa	HMCb	AMCa	AMCb
$S_0 = 80$	0.01	1.1302	1.1302	1.1293	10000	1.08±0.09	1.11±0.09	1.00±0.09	1.08±0.09
	0.005	1.1293	1.1294		50000	1.12±0.04	1.17±0.04	1.07±0.04	1.10±0.04
	0.0025	1.1291	1.1292		100000	1.14±0.03	1.14±0.03	1.13±0.03	1.13±0.03
	0.00125	1.1291	1.1292		200000	1.13±0.02	1.14±0.02	1.11±0.02	1.12±0.02
$S_0 = 90$	0.01	3.3331	3.3312	3.3284	10000	3.27±0.17	3.27±0.17	3.19±0.16	3.22±0.16
	0.005	3.3315	3.3301		50000	3.32±0.08	3.40±0.08	3.24±0.07	3.26±0.0
	0.0025	3.3311	3.3298		100000	3.34±0.05	3.34±0.05	3.32±0.05	3.33±0.05
	0.00125	3.3310	3.3297		200000	3.32±0.04	3.35±0.04	3.28±0.04	3.31±0.04
$S_0 = 100$	0.01	7.5245	7.5239	7.5210	10000	7.46±0.25	7.46±0.25	7.37±0.24	7.36±0.25
	0.005	7.5236	7.5224		50000	7.53±0.11	7.62±0.11	7.40±0.11	7.43±0.11
	0.0025	7.5231	7.5221		100000	7.54±0.08	7.52±0.08	7.53±0.08	7.52±0.08
	0.00125	7.5230	7.5220		200000	7.50±0.06	7.54±0.06	7.46±0.06	7.50±0.06
$S_0 = 110$	0.01	13.6943	13.6940	13.6923	10000	13.69±0.34	13.69±0.34	13.52±0.33	13.48±0.33
	0.005	13.6923	13.6924		50000	13.71±0.15	13.81±0.15	13.55±0.15	13.58±0.15
	0.0025	13.6918	13.6921		100000	13.72±0.11	13.69±0.11	13.67±0.11	13.70±0.11
	0.00125	13.6917	13.6920		200000	13.64±0.08	13.71±0.08	13.63±0.07	13.69±0.08
$S_0 = 120$	0.01	21.3173	21.3185	21.3174	10000	21.40±0.41	21.40±0.41	21.08±0.40	21.03±0.41
	0.005	21.3156	21.3168		50000	21.35±0.18	21.46±0.19	21.17±0.18	21.21±0.18
	0.0025	21.3152	21.3164		100000	21.36±0.13	21.32±0.13	21.29±0.13	21.33±0.13
	0.00125	21.3152	21.3163		200000	21.25±0.09	21.33±0.09	21.26±0.09	21.33±0.09

(b)

$\rho = 0.5$	Δy	HTFDa	HTFDb	CF	N_{MC}	HMCa	HMCb	AMCa	AMCb
$S_0 = 80$	0.01	1.4732	1.4751	1.4760	10000	1.42±0.12	1.40±0.12	1.37±0.12	1.35±0.12
	0.005	1.4724	1.4744		50000	1.49±0.06	1.47±0.05	1.40±0.05	1.42±0.05
	0.0025	1.4723	1.4742		100000	1.48±0.04	1.46±0.04	1.46±0.04	1.49±0.04
	0.00125	1.4722	1.4741		200000	1.47±0.03	1.48±0.03	1.48±0.03	1.48±0.03
$S_0 = 90$	0.01	3.6849	3.6859	3.6862	10000	3.63±0.19	3.63±0.19	3.48±0.19	3.49±0.19
	0.005	3.6836	3.6849		50000	3.70±0.09	3.70±0.09	3.57±0.09	3.60±0.09
	0.0025	3.6832	3.6847		100000	3.67±0.06	3.67±0.06	3.66±0.06	3.71±0.06
	0.00125	3.6832	3.6847		200000	3.66±0.04	3.70±0.04	3.69±0.04	3.68±0.04
$S_0 = 100$	0.01	7.6247	7.6245	7.6223	10000	7.58±0.28	7.58±0.28	7.35±0.28	7.36±0.27
	0.005	7.6238	7.6232		50000	7.66±0.13	7.65±0.13	7.47±0.12	7.52±0.12
	0.0025	7.6234	7.6229		100000	7.61±0.09	7.59±0.09	7.58±0.09	7.66±0.09
	0.00125	7.6233	7.6228		200000	7.58±0.06	7.64±0.06	7.62±0.06	7.61±0.06
$S_0 = 110$	0.01	13.4863	13.4835	13.4791	10000	13.48±0.36	13.48±0.36	13.21±0.36	13.19±0.36
	0.005	13.4842	13.4818		50000	13.55±0.17	13.49±0.16	13.27±0.16	13.35±0.16
	0.0025	13.4837	13.4814		100000	13.47±0.12	13.41±0.12	13.44±0.12	13.54±0.12
	0.00125	13.4836	13.4813		200000	13.42±0.08	13.49±0.08	13.47±0.08	13.48±0.08
$S_0 = 120$	0.01	20.9678	20.9661	20.9616	10000	21.04±0.44	21.04±0.44	20.67±0.44	20.64±0.43
	0.005	20.9659	20.9642		50000	21.05±0.20	20.98±0.20	20.71±0.20	20.81±0.20
	0.0025	20.9655	20.9636		100000	20.96±0.14	20.87±0.14	20.92±0.14	21.04±0.14
	0.00125	20.9654	20.9635		200000	20.88±0.10	20.96±0.10	20.97±0.10	20.98±0.10

Table 1: *Standard Bates model. Prices of European call options. Test parameters: $K = 100$, $T = 0.5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta_V = 0.04$, $\kappa_V = 2$, $\sigma_V = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5, 0.5$.*

(a)

$\rho = -0.5$	Δy	HTFDa	HTFDb	PSOR	MOL	N_{MC}	HMCLSa	HMCL Sb	AMCLSa	AMCL Sb
$S_0 = 80$	0.01	1.1365	1.1365	1.1359	1.1363	10000	1.03±0.08	1.14±0.09	1.06±0.09	1.03±0.09
	0.005	1.1356	1.1358			50000	1.19±0.04	1.14±0.04	1.18±0.04	1.12±0.04
	0.0025	1.1354	1.1356			100000	1.15±0.03	1.13±0.03	1.13±0.03	1.13±0.03
	0.00125	1.1353	1.1355			200000	1.14±0.02	1.14±0.02	1.14±0.02	1.14±0.02
$S_0 = 90$	0.01	3.3579	3.3563	3.3532	3.3530	10000	3.39±0.15	3.44±0.16	3.38±0.15	3.48±0.16
	0.005	3.3564	3.3551			50000	3.46±0.07	3.33±0.07	3.46±0.07	3.32±0.07
	0.0025	3.3560	3.3548			100000	3.35±0.05	3.35±0.05	3.33±0.05	3.36±0.05
	0.00125	3.3559	3.3547			200000	3.35±0.03	3.33±0.03	3.35±0.03	3.34±0.03
$S_0 = 100$	0.01	7.6010	7.6006	7.5970	7.5959	10000	7.68±0.23	7.88±0.24	7.63±0.23	7.80±0.24
	0.005	7.6001	7.5992			50000	7.75±0.11	7.59±0.10	7.76±0.10	7.53±0.10
	0.0025	7.5997	7.5989			100000	7.56±0.07	7.61±0.07	7.56±0.07	7.61±0.07
	0.00125	7.5996	7.5989			200000	7.58±0.05	7.55±0.05	7.58±0.05	7.57±0.05
$S_0 = 110$	0.01	13.8853	13.8854	13.8830	13.8827	10000	13.90±0.29	14.28±0.30	13.84±0.29	14.10±0.29
	0.005	13.8836	13.8842			50000	14.05±0.13	13.89±0.12	14.07±0.13	13.86±0.12
	0.0025	13.8832	13.8839			100000	13.80±0.09	13.91±0.09	13.84±0.09	13.89±0.09
	0.00125	13.8831	13.8838			200000	13.86±0.06	13.84±0.06	13.87±0.06	13.83±0.06
$S_0 = 120$	0.01	21.7180	21.7199	21.7186	21.7191	10000	21.83±0.34	22.07±0.33	21.71±0.30	22.04±0.34
	0.005	21.7168	21.7187			50000	21.91±0.15	21.76±0.13	21.90±0.15	21.72±0.13
	0.0025	21.7166	21.7184			100000	21.59±0.10	21.78±0.10	21.64±0.10	21.72±0.10
	0.00125	21.7165	21.7183			200000	21.68±0.07	21.65±0.07	21.68±0.07	21.67±0.07

(b)

$\rho = 0.5$	Δy	HTFDa	HTFDb	PSOR	MOL	N_{MC}	HMCLSa	HMCL Sb	AMCLSa	AMCL Sb
$S_0 = 80$	0.01	1.4817	1.4837	1.4843	1.4848	10000	1.32±0.11	1.03±0.09	1.51±0.13	0.66±0.08
	0.005	1.4809	1.4830			50000	1.51±0.05	1.31±0.05	1.54±0.05	1.47±0.05
	0.0025	1.4807	1.4828			100000	1.50±0.04	1.50±0.04	1.51±0.04	1.48±0.04
	0.00125	1.4807	1.4828			200000	1.50±0.03	1.49±0.02	1.49±0.03	1.47±0.02
$S_0 = 90$	0.01	3.7134	3.7148	3.7145	3.7146	10000	3.83±0.19	3.79±0.17	3.89±0.19	3.95±0.19
	0.005	3.7121	3.7139			50000	3.81±0.08	3.70±0.08	3.84±0.08	3.69±0.08
	0.0025	3.7118	3.7137			100000	3.69±0.06	3.75±0.06	3.72±0.06	3.70±0.06
	0.00125	3.7118	3.7137			200000	3.70±0.04	3.71±0.04	3.72±0.04	3.70±0.04
$S_0 = 100$	0.01	7.7044	7.7051	7.7027	7.7018	10000	7.74±0.26	7.85±0.25	7.96±0.26	7.99±0.26
	0.005	7.7036	7.7039			50000	7.85±0.12	7.68±0.11	7.87±0.12	7.68±0.11
	0.0025	7.7033	7.7036			100000	7.66±0.08	7.75±0.08	7.65±0.08	7.73±0.08
	0.00125	7.7032	7.7036			200000	7.69±0.06	7.67±0.05	7.68±0.06	7.69±0.05
$S_0 = 110$	0.01	13.6770	13.6756	13.6722	13.6715	10000	13.57±0.32	13.98±0.31	13.88±0.32	14.12±0.33
	0.005	13.6752	13.6742			50000	13.83±0.14	13.67±0.13	13.89±0.14	13.64±0.13
	0.0025	13.6747	13.6739			100000	13.56±0.09	13.74±0.10	13.58±0.10	13.71±0.10
	0.00125	13.6747	13.6738			200000	13.65±0.07	13.65±0.07	13.64±0.07	13.64±0.07
$S_0 = 120$	0.01	21.3668	21.3671	21.3653	21.3657	10000	21.45±0.32	21.60±0.35	21.39±0.33	21.84±0.34
	0.005	21.3655	21.3658			50000	21.54±0.15	21.40±0.14	21.61±0.16	21.40±0.13
	0.0025	21.3653	21.3655			100000	21.26±0.10	21.43±0.10	21.27±0.10	21.38±0.10
	0.00125	21.3652	21.3653			200000	21.31±0.07	21.33±0.07	21.31±0.07	21.31±0.07

Table 2: *Standard Bates model. Prices of American call options. Test parameters: $K = 100$, $T = 0.5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta_V = 0.04$, $\kappa_V = 2$, $\sigma_V = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5, 0.5$.*

Δy	HTFDa	HTDFb	N_{MC}	HMCa	HMCb	AMCa	AMCb	CF
0.01	0.09	0.34	10000	0.007	0.16	0.16	0.30	0.001
0.005	0.18	0.72	50000	0.36	0.72	0.79	1.51	
0.0025	0.46	1.62	100000	0.71	1.44	1.57	3.12	
0.00125	0.84	3.53	200000	1.45	2.95	3.14	6.17	

Table 3: *Standard Bates model. Computational times (in seconds) for European call options in Table 1 for $S_0 = 100$, $\rho = -0.5$.*

Δy	HTFDa	HTDFb	N_{MC}	HMCLSa	HMCLSb	AMCLSa	AMCLSb
0.01	0.10	0.37	10000	0.09	0.23	0.20	0.45
0.005	0.19	0.77	50000	0.47	1.11	1.01	2.25
0.0025	0.48	1.77	100000	1.07	2.25	2.01	4.57
0.00125	0.95	3.61	200000	1.94	4.55	4.05	8.98

Table 4: *Standard Bates model. Computational times (in seconds) for American call options in Table 2 for $S_0 = 100$, $\rho = -0.5$.*

N	$S_0 = 80$	$S_0 = 90$	$S_0 = 100$	$S_0 = 110$	$S_0 = 120$
200	1.919250	1.961063	1.894156	2.299666	2.109026
400	2.172836	2.209762	2.556021	1.673541	1.996332
800	1.544849	1.851932	1.463712	2.935697	2.106880

Table 5: *Standard Bates model. HTFDb-ratio (6.4) for the price of American call options as the starting point S_0 varies with fixed space step $\Delta y = 0.0025$. Test parameters: $T = 0.5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta = 0.04$, $\kappa = 2$, $\sigma = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5$.*

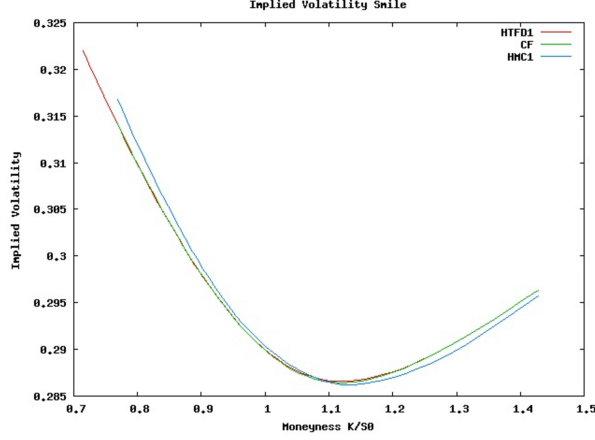


Figure 2: *Standard Bates model. Moneyness vs implied volatility for European call options with $N_t = 50$. The red line refers to **HTFDa** with $\Delta y = 0.005$, the blue line refers to **HMCa** with $N_{MC} = 50000$ and the green line refers to the benchmark values **CF**. The accuracy of our hybrid methods is evident.*

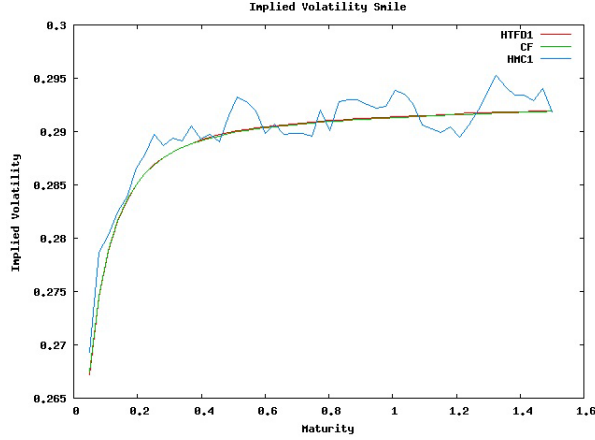


Figure 3: *Standard Bates model. Maturity vs implied volatility for European call options with $N_t = 50$. The red line refers to **HTFDa** with $\Delta y = 0.005$, the blue line refers to **HMCa** with $N_{MC} = 50000$, and the green line refers to the benchmark values **CF**. Also in this case the accuracy of our hybrid methods is evident.*

6.2.2 Options with large maturity in the standard Bates model

In order to verify the robustness of the proposed algorithms we consider experiments when the Feller condition $2\kappa_V\theta_V \geq \sigma_V^2$ is not fulfilled for the CIR volatility process. We additionally stress our tests by considering large maturities. For this purpose we consider the parameters from Chiarella *et al.* [16] already used in Section 6.2.1 with $\rho = -0.5$, except for the maturity and the vol-vol, which are modified as follows: $T = 5$ and $\sigma_V = 0.7$ respectively.

Table 6 reports European call option prices, which are compared with the true values (**CF**). In Table 7 we provide results for American call option prices. The settings for the experiments **HTFDa-b**, **HMCa-b** and **AMCa-b** are the same as described at the beginning of Section 6.2.1. The settings for the experiments in the American case **HMCLSa-b** and **AMCLSa-b** are changed

HMCLSa and **AMCLSa**: 20 exercise dates, $N_t = 100$ and $N_{MC} = 10000, 50000, 100000, 200000$;

HMCL Sb and **AMCL Sb**: 40 exercise dates, $N_t = 200$ and $N_{MC} = 10000, 50000, 100000, 200000$.

In the American case the benchmark values **B-AMC** are obtained by the Longstaff-Schwartz [33] Monte Carlo algorithm with 300 exercise dates, combined with the accurate third-order Alfonsi method with 3000 discretization time steps and 1 million iterations.

The numerical results suggest that large maturities bring to a slight loss of accuracy for **HTFD** and **HMC**, even if both methods provide a satisfactory approximation of the true option prices, being in turn mostly compatible with the results from the Alfonsi Monte Carlo method. It is worth noticing that for long maturity $T = 5$ we have developed experiments with the same number of steps both in time (N_t) and space step (Δy) as for $T = 0.5$. So, the numerical experiments are not slower, and it is clear that one could achieve a better accuracy for larger values of N_t .

$\rho = -0.5$	Δy	HTFDa	HTFDb	CF	N_{MC}	HMCa	HMCb	AMCa	AMCb
$S_0 = 80$	0.01	9.0085	8.9457	8.9262	10000	9.21±0.55	9.09±0.55	8.69±0.53	8.56±0.51
	0.0050	9.0032	8.9405		50000	9.13±0.25	8.92±0.24	8.81±0.24	9.04±0.24
	0.0025	9.0020	8.9392		100000	9.01±0.17	8.81±0.17	8.92±0.17	8.88±0.17
	0.00125	9.0016	8.9389		200000	8.99±0.12	8.92±0.12	8.95±0.12	8.90±0.12
$S_0 = 90$	0.01	12.7405	12.6520	12.6257	10000	12.95±0.67	12.95±0.67	12.29±0.65	12.15±0.6
	0.0050	12.7342	12.6458		50000	12.87±0.30	12.64±0.29	12.49±0.29	12.76±0.3
	0.0025	12.7327	12.6442		100000	12.72±0.21	12.50±0.21	12.63±0.21	12.58±0.21
	0.00125	12.7323	12.6438		200000	12.71±0.15	12.61±0.15	12.66±0.15	12.61±0.15
$S_0 = 100$	0.01	17.0324	16.9176	16.8855	10000	17.24±0.80	17.24±0.80	16.43±0.77	16.29±0.75
	0.0050	17.0254	16.9106		50000	17.18±0.36	16.91±0.35	16.73±0.35	17.03±0.35
	0.0025	17.0237	16.9089		100000	17.00±0.25	16.74±0.25	16.91±0.25	16.84±0.25
	0.00125	17.0232	16.9084		200000	16.99±0.18	16.86±0.18	16.94±0.18	16.88±0.18
$S_0 = 110$	0.01	21.8149	21.6741	21.6364	10000	22.04±0.93	22.04±0.93	21.06±0.93	20.91±0.88
	0.0050	21.8067	21.6659		50000	21.96±0.42	21.67±0.41	21.43±0.41	21.82±0.41
	0.0025	21.8047	21.6639		100000	21.76±0.29	21.47±0.29	21.69±0.29	21.59±0.29
	0.00125	21.8042	21.6634		200000	21.76±0.21	21.59±0.20	21.70±0.20	21.63±0.20
$S_0 = 120$	0.01	27.0196	26.8539	26.8121	10000	27.26±1.05	27.26±1.05	26.12±1.03	25.94±1.01
	0.0050	27.0108	26.8452		50000	27.17±0.47	26.86±0.46	26.56±0.46	27.02±0.47
	0.0025	27.0086	26.8430		100000	26.94±0.33	26.63±0.33	26.89±0.33	26.78±0.33
	0.00125	27.0081	26.8425		200000	26.95±0.23	26.75±0.23	26.89±0.23	26.81±0.23

Table 6: *Standard Bates model. Prices of European call options. Test parameters: $K = 100$, $T = 5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta_V = 0.04$, $\kappa_V = 2$, $\sigma_V = 0.7$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5$. Case $2\kappa_V\theta_V < \sigma_V^2$.*

6.2.3 Bates model with stochastic interest rate

We consider now the case of Bates model associated with the Vasicek model for the stochastic interest rate. For the Bates model we consider the parameters from Chiarella *et al.* [16] already used in Section 6.2.3. Moreover, for the interest rate parameter we fix the following parameters: initial interest rate $r_0 = 0.03$, speed of mean-reversion $\kappa_r = 1$, interest rate volatility $\sigma_r = 0.2$; time-varying long-term mean $\theta_r(t)$ fitting the theoretical bond prices to the yield curve observed on the market, here set as $P_r(0, T) = e^{-0.03T}$. We study the cases $\rho_1 = \rho_{SV} = -0.5$ and $\rho_2 = \rho_{Sr} = -0.5, 0.5$. No correlation is assumed to exist between r and V . We consider the mesh grid $\Delta y = 0.02, 0.01, 0.005, 0.0025$, the case $\Delta y = 0.00125$ being removed because it requires huge computational times. The numerical results are labeled **HTFDa-b**, **HMCa-b**, **AMCa-b**, **HMCLSa-b**, **AMCLSa-b**, their settings being given at the beginning of Section 6.2.1.

$\rho = -0.5$	Δy	HTFDa	HTFDb	B-AMC	N_{MC}	HMCLSa	HMCLSc	AMCLSa	AMCLSc
$S_0 = 80$	0.01	9.8335	9.7978	9.7907 ± 0.04	10000	10.15 ± 0.46	10.20 ± 0.46	10.47 ± 0.47	9.80 ± 0.42
	0.0050	9.8283	9.7927		50000	9.93 ± 0.20	9.86 ± 0.20	9.89 ± 0.19	9.78 ± 0.19
	0.0025	9.8271	9.7914		100000	9.76 ± 0.14	9.69 ± 0.13	9.74 ± 0.14	9.76 ± 0.13
	0.00125	9.8267	9.7911		200000	9.79 ± 0.10	9.70 ± 0.09	9.73 ± 0.10	9.72 ± 0.09
$S_0 = 90$	0.01	14.0801	14.0318	14.0030 ± 0.05	10000	14.58 ± 0.56	14.46 ± 0.55	14.94 ± 0.58	14.08 ± 0.51
	0.0050	14.0741	14.0258		50000	14.13 ± 0.24	14.14 ± 0.24	14.19 ± 0.23	14.12 ± 0.23
	0.0025	14.0726	14.0244		100000	13.98 ± 0.16	13.87 ± 0.16	13.94 ± 0.16	13.89 ± 0.16
	0.00125	14.0722	14.0240		200000	13.93 ± 0.12	13.91 ± 0.11	13.94 ± 0.12	13.96 ± 0.11
$S_0 = 100$	0.01	19.0658	19.0075	18.9632 ± 0.05	10000	19.59 ± 0.66	19.44 ± 0.63	19.88 ± 0.66	19.13 ± 0.59
	0.0050	19.0594	19.0011		50000	19.10 ± 0.27	19.06 ± 0.27	19.26 ± 0.26	19.01 ± 0.26
	0.0025	19.0578	18.9995		100000	18.92 ± 0.19	18.88 ± 0.18	18.85 ± 0.19	18.90 ± 0.18
	0.00125	19.0574	18.9991		200000	18.80 ± 0.13	18.84 ± 0.13	18.85 ± 0.13	18.92 ± 0.13
$S_0 = 110$	0.01	24.7434	24.6788	24.6289 ± 0.06	10000	25.02 ± 0.74	24.84 ± 0.72	25.32 ± 0.72	24.78 ± 0.67
	0.0050	24.7364	24.6719		50000	24.79 ± 0.30	24.57 ± 0.29	24.94 ± 0.29	24.72 ± 0.29
	0.0025	24.7347	24.6701		100000	24.53 ± 0.21	24.47 ± 0.20	24.50 ± 0.21	24.51 ± 0.20
	0.00125	24.7343	24.6697		200000	24.42 ± 0.14	24.45 ± 0.14	24.50 ± 0.15	24.53 ± 0.14
$S_0 = 120$	0.01	31.0646	30.9983	30.9052 ± 0.07	10000	30.88 ± 0.74	31.15 ± 0.75	31.18 ± 0.74	31.04 ± 0.71
	0.0050	31.0577	30.9914		50000	31.10 ± 0.32	30.94 ± 0.31	31.32 ± 0.32	30.98 ± 0.32
	0.0025	31.0559	30.9896		100000	30.89 ± 0.23	30.72 ± 0.22	30.70 ± 0.22	30.72 ± 0.22
	0.00125	31.0555	30.9892		200000	30.72 ± 0.16	30.73 ± 0.16	30.77 ± 0.16	30.89 ± 0.15

Table 7: *Standard Bates model. Prices of American call options. Test parameters: $K = 100$, $T = 5$, $r = 0.03$, $\eta = 0.05$, $V_0 = 0.04$, $\theta_V = 0.04$, $\kappa_V = 2$, $\sigma_V = 0.7$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho = -0.5$. Case $2\kappa_V\theta_V < \sigma_V^2$.*

When the interest rate is assumed to be stochastic, no references are available in the literature. Therefore, we propose benchmark values obtained by using a Monte Carlo method in which the CIR paths are simulated through the accurate third-order Alfonsi [2] discretization scheme and the interest rate paths are generated by an exact scheme. For these benchmark values, called **B-AMC**, the number of Monte Carlo iterations and of the discretization time steps are set as $N_{MC} = 10^6$ and $N_t = 300$ respectively. In the American case, **B-AMC** is evaluated through the Longstaff-Schwartz [33] algorithm with 20 exercise dates. All Monte Carlo results report the 95% confidence intervals.

European and American call option prices are given in tables 8 and 9 respectively. Tables 10 and 11 refer to the computational time cost (in seconds) of the different algorithms in the European Call case and American Call case respectively. The numerical results confirm the good numerical behavior of **HTFD** and **HMC** in the Bates-Hull-White model as well.

(a)

$\rho_{Sr} = -0.5$	Δy	HTFDa	HTFDb	B-AMC	N_{MC}	HMCa	HMCb	AMCa	AMCb
$S_0 = 80$	0.02	1.0169	1.0079	1.0153 ± 0.01	10000	1.00 ± 0.09	0.96 ± 0.09	1.00 ± 0.09	1.06 ± 0.10
	0.01	1.0201	1.0188		50000	1.02 ± 0.04	0.97 ± 0.04	0.98 ± 0.04	1.01 ± 0.04
	0.0050	1.0199	1.0194		100000	1.00 ± 0.03	1.00 ± 0.03	1.01 ± 0.03	1.03 ± 0.03
	0.0025	1.0197	1.0193		200000	1.01 ± 0.02	1.01 ± 0.02	1.02 ± 0.02	1.00 ± 0.02
$S_0 = 90$	0.01	3.1172	3.1032	3.1008 ± 0.02	10000	3.05 ± 0.16	3.05 ± 0.16	3.07 ± 0.16	3.14 ± 0.17
	0.01	3.1186	3.1137		50000	3.10 ± 0.07	3.03 ± 0.07	3.02 ± 0.07	3.09 ± 0.07
	0.0050	3.1174	3.1135		100000	3.07 ± 0.05	3.08 ± 0.05	3.09 ± 0.05	3.14 ± 0.05
	0.0025	3.1174	3.1136		200000	3.09 ± 0.04	3.10 ± 0.04	3.11 ± 0.04	3.08 ± 0.04
$S_0 = 100$	0.02	7.2528	7.2472	7.2315 ± 0.02	10000	7.17 ± 0.24	7.17 ± 0.24	7.20 ± 0.24	7.24 ± 0.25
	0.01	7.2528	7.2479		50000	7.21 ± 0.11	7.18 ± 0.11	7.12 ± 0.11	7.21 ± 0.11
	0.0050	7.2528	7.2480		100000	7.18 ± 0.08	7.24 ± 0.08	7.20 ± 0.08	7.27 ± 0.08
	0.0025	7.2528	7.2480		200000	7.22 ± 0.05	7.25 ± 0.05	7.24 ± 0.05	7.20 ± 0.05
$S_0 = 110$	0.02	13.4553	13.4565	13.4256 ± 0.03	10000	13.30 ± 0.32	13.30 ± 0.32	13.41 ± 0.33	13.39 ± 0.33
	0.01	13.4465	13.4440		50000	13.37 ± 0.15	13.40 ± 0.15	13.27 ± 0.15	13.38 ± 0.15
	0.0050	13.4435	13.4407		100000	13.35 ± 0.10	13.46 ± 0.10	13.38 ± 0.10	13.48 ± 0.10
	0.0025	13.4432	13.4404		200000	13.40 ± 0.07	13.47 ± 0.07	13.43 ± 0.07	13.39 ± 0.07
$S_0 = 120$	0.02	21.1320	21.1356	21.1070 ± 0.04	10000	20.89 ± 0.40	20.89 ± 0.40	21.08 ± 0.40	20.99 ± 0.41
	0.01	21.1243	21.1239		50000	21.03 ± 0.18	21.09 ± 0.18	20.92 ± 0.18	21.03 ± 0.18
	0.0050	21.1222	21.1214		100000	21.01 ± 0.13	21.17 ± 0.13	21.04 ± 0.13	21.17 ± 0.13
	0.0025	21.1215	21.1207		200000	21.06 ± 0.09	21.16 ± 0.09	21.12 ± 0.09	21.06 ± 0.09

(b)

$\rho_{Sr} = 0.5$	Δy	HTFDa	HTFDb	B-AMC	N_{MC}	HMCa	HMCb	AMCa	AMCb
$S_0 = 80$	0.02	1.3459	1.3379	1.3446 ± 0.01	10000	1.29 ± 0.11	1.28 ± 0.11	1.32 ± 0.10	1.41 ± 0.11
	0.01	1.3482	1.3471		50000	1.34 ± 0.05	1.30 ± 0.05	1.32 ± 0.05	1.35 ± 0.05
	0.0050	1.3479	1.3475		100000	1.32 ± 0.03	1.31 ± 0.03	1.34 ± 0.03	1.34 ± 0.03
	0.0025	1.3477	1.3473		200000	1.33 ± 0.02	1.34 ± 0.02	1.35 ± 0.02	1.32 ± 0.02
$S_0 = 90$	0.01	3.7320	3.7233	3.7263 ± 0.02	10000	3.62 ± 0.18	3.62 ± 0.18	3.64 ± 0.18	3.76 ± 0.19
	0.01	3.7323	3.7304		50000	3.69 ± 0.08	3.65 ± 0.08	3.64 ± 0.18	3.76 ± 0.19
	0.0050	3.7311	3.7298		100000	3.66 ± 0.06	3.68 ± 0.06	3.71 ± 0.06	3.73 ± 0.06
	0.0025	3.7311	3.7299		200000	3.69 ± 0.04	3.72 ± 0.04	3.73 ± 0.04	3.68 ± 0.04
$S_0 = 100$	0.02	8.0100	8.0073	8.0069 ± 0.03	10000	7.83 ± 0.26	7.83 ± 0.26	7.82 ± 0.26	8.00 ± 0.27
	0.01	8.0112	8.0102		50000	7.92 ± 0.12	7.93 ± 0.12	7.93 ± 0.12	7.97 ± 0.12
	0.0050	8.0114	8.0107		100000	7.91 ± 0.08	7.97 ± 0.08	7.99 ± 0.08	8.02 ± 0.08
	0.0025	8.0114	8.0107		200000	7.95 ± 0.06	8.02 ± 0.06	8.00 ± 0.06	7.95 ± 0.06
$S_0 = 110$	0.02	14.1482	14.1505	14.1323 ± 0.03	10000	13.89 ± 0.35	13.89 ± 0.35	13.88 ± 0.35	14.07 ± 0.36
	0.01	14.1413	14.1414		50000	14.01 ± 0.16	14.05 ± 0.16	14.03 ± 0.16	14.09 ± 0.16
	0.0050	14.1388	14.1388		100000	14.01 ± 0.11	14.10 ± 0.11	14.12 ± 0.11	14.14 ± 0.11
	0.0025	14.1386	14.1386		200000	14.06 ± 0.08	14.17 ± 0.08	14.13 ± 0.08	14.07 ± 0.08
$S_0 = 120$	0.02	21.6737	21.6772	21.6501 ± 0.04	10000	21.37 ± 0.42	21.37 ± 0.42	21.35 ± 0.42	21.51 ± 0.43
	0.01	21.6670	21.6674		50000	21.50 ± 0.19	21.55 ± 0.19	21.52 ± 0.19	21.60 ± 0.19
	0.0050	21.6651	21.6653		100000	21.52 ± 0.13	21.63 ± 0.13	21.64 ± 0.13	21.68 ± 0.14
	0.0025	21.6645	21.6646		200000	21.57 ± 0.10	21.71 ± 0.10	21.65 ± 0.10	21.58 ± 0.09

Table 8: *Bates-Hull-White model. Prices of European call options. Test parameters: $K = 100$, $T = 0.5$, $\eta = 0.05$, $r_0 = 0.03$, $\kappa_r = 1$, $\sigma_r = 0.2$, $V_0 = 0.04$, $\theta_V = 0.04$, $\kappa_V = 2$, $\sigma_V = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho_{SV} = -0.5$, $\rho_{Sr} = -0.5$, 0.5 .*

(a)

$\rho_{Sr} = -0.5$	Δy	HTFDa	HTFDb	B-AMC	N_{MC}	HMCLSa	HMCLSB	AMCLSa	AMCLSB
$S_0 = 80$	0.02	1.0561	1.0470	1.0544 ± 0.01	10000	0.76 ± 0.07	0.56 ± 0.06	0.95 ± 0.08	0.82 ± 0.08
	0.01	1.0598	1.0588		50000	1.08 ± 0.04	0.91 ± 0.04	1.01 ± 0.04	0.96 ± 0.04
	0.0050	1.0597	1.0596		100000	1.07 ± 0.03	1.03 ± 0.03	1.07 ± 0.03	1.04 ± 0.03
	0.0025	1.0596	1.0595		200000	1.05 ± 0.02	1.04 ± 0.02	1.07 ± 0.02	1.05 ± 0.02
$S_0 = 90$	0.01	3.2511	3.2364	3.2273 ± 0.01	10000	3.28 ± 0.15	3.39 ± 0.16	3.35 ± 0.16	3.07 ± 0.15
	0.01	3.2537	3.2493		50000	3.33 ± 0.07	3.21 ± 0.07	3.25 ± 0.07	3.30 ± 0.07
	0.0050	3.2528	3.2494		100000	3.23 ± 0.05	3.24 ± 0.05	3.27 ± 0.05	3.25 ± 0.05
	0.0025	3.2528	3.2495		200000	3.22 ± 0.03	3.23 ± 0.03	3.25 ± 0.03	3.24 ± 0.03
$S_0 = 100$	0.02	7.6012	7.5952	7.5589 ± 0.02	10000	7.64 ± 0.22	7.99 ± 0.23	7.80 ± 0.23	7.68 ± 0.22
	0.01	7.6020	7.5976		50000	7.72 ± 0.10	7.58 ± 0.09	7.61 ± 0.10	7.65 ± 0.10
	0.0050	7.6022	7.5980		100000	7.54 ± 0.07	7.62 ± 0.07	7.61 ± 0.07	7.54 ± 0.07
	0.0025	7.6022	7.5980		200000	7.54 ± 0.05	7.54 ± 0.05	7.56 ± 0.05	7.60 ± 0.05
$S_0 = 110$	0.02	14.1510	14.1524	14.0909 ± 0.03	10000	14.22 ± 0.28	14.61 ± 0.29	14.35 ± 0.29	14.07 ± 0.28
	0.01	14.1443	14.1425		50000	14.25 ± 0.13	14.11 ± 0.12	14.16 ± 0.12	14.17 ± 0.13
	0.0050	14.1420	14.1401		100000	14.03 ± 0.09	14.18 ± 0.09	14.10 ± 0.09	14.06 ± 0.09
	0.0025	14.1419	14.1399		200000	14.05 ± 0.06	14.04 ± 0.06	14.07 ± 0.06	14.13 ± 0.06
$S_0 = 120$	0.02	22.2466	22.2505	22.1736 ± 0.03	10000	22.38 ± 0.32	22.84 ± 0.33	22.46 ± 0.32	22.15 ± 0.32
	0.01	22.2412	22.2419		50000	22.35 ± 0.15	22.27 ± 0.14	22.24 ± 0.14	22.28 ± 0.14
	0.0050	22.2398	22.2402		100000	22.12 ± 0.10	22.27 ± 0.10	22.19 ± 0.10	22.17 ± 0.10
	0.0025	22.2394	22.2397		100000	22.12 ± 0.10	22.27 ± 0.10	22.19 ± 0.10	22.17 ± 0.10

(b)

$\rho_{Sr} = 0.5$	Δy	HTFDa	HTFDb	B-AMC	N_{MC}	HMCLSa	HMCLSB	AMCLSa	AMCLSB
$S_0 = 80$	0.02	1.3551	1.3470	1.3559 ± 0.01	10000	1.18 ± 0.09	1.29 ± 0.10	1.12 ± 0.09	0.80 ± 0.08
	0.01	1.3576	1.3566		50000	1.35 ± 0.05	1.17 ± 0.04	1.33 ± 0.05	1.25 ± 0.05
	0.0050	1.3573	1.3570		100000	1.33 ± 0.03	1.30 ± 0.03	1.33 ± 0.03	1.27 ± 0.03
	0.0025	1.3571	1.3569		200000	1.35 ± 0.02	1.31 ± 0.02	1.38 ± 0.02	1.34 ± 0.02
$S_0 = 90$	0.01	3.7696	3.7606	3.7633 ± 0.02	10000	3.72 ± 0.17	3.78 ± 0.17	3.82 ± 0.18	3.72 ± 0.17
	0.01	3.7705	3.7688		50000	3.86 ± 0.08	3.71 ± 0.08	3.80 ± 0.08	3.81 ± 0.08
	0.0050	3.7694	3.7685		100000	3.75 ± 0.06	3.74 ± 0.05	3.76 ± 0.05	3.74 ± 0.05
	0.0025	3.7694	3.7686		200000	3.75 ± 0.04	3.74 ± 0.04	3.80 ± 0.04	3.79 ± 0.04
$S_0 = 100$	0.02	8.1285	8.1249	8.1122 ± 0.03	10000	8.12 ± 0.24	8.52 ± 0.26	8.25 ± 0.26	8.15 ± 0.25
	0.01	8.1308	8.1301		50000	8.25 ± 0.11	8.08 ± 0.11	8.15 ± 0.11	8.18 ± 0.11
	0.0050	8.1311	8.1308		100000	8.07 ± 0.08	8.16 ± 0.08	8.11 ± 0.08	8.10 ± 0.08
	0.0025	8.1312	8.1309		200000	8.08 ± 0.06	8.07 ± 0.06	8.14 ± 0.06	8.16 ± 0.06
$S_0 = 110$	0.02	14.4455	14.4468	14.3884 ± 0.03	10000	14.48 ± 0.32	14.84 ± 0.33	14.43 ± 0.32	14.51 ± 0.32
	0.01	14.4409	14.4414		50000	14.60 ± 0.15	14.40 ± 0.14	14.45 ± 0.14	14.47 ± 0.14
	0.0050	14.4389	14.4395		100000	14.34 ± 0.10	14.47 ± 0.10	14.39 ± 0.10	14.38 ± 0.10
	0.0025	14.4388	14.4394		200000	14.35 ± 0.07	14.37 ± 0.07	14.38 ± 0.07	14.48 ± 0.07
$S_0 = 120$	0.02	22.2859	22.2893	22.2039 ± 0.04	10000	22.23 ± 0.36	22.87 ± 0.39	22.45 ± 0.36	22.29 ± 0.35
	0.01	22.2815	22.2827		50000	22.50 ± 0.17	22.29 ± 0.16	22.27 ± 0.16	22.28 ± 0.16
	0.0050	22.2802	22.2813		100000	22.17 ± 0.12	22.31 ± 0.12	22.24 ± 0.12	22.22 ± 0.12
	0.0025	22.2798	22.2808		200000	22.17 ± 0.08	22.17 ± 0.08	22.17 ± 0.08	22.32 ± 0.08

Table 9: *Bates-Hull-White model. Prices of American call options. Test parameters: $K = 100$, $T = 0.5$, $\eta = 0.05$, $r_0 = 0.03$, $\kappa_r = 1$, $\sigma_r = 0.2$, $V_0 = 0.04$, $\theta_V = 0.04$, $\kappa_V = 2$, $\sigma_V = 0.4$, $\lambda = 5$, $\gamma = 0$, $\delta = 0.1$, $\rho_{SV} = -0.5$, $\rho_{Sr} = -0.5, 0.5$.*

Δy	HTFDa	HTDFb	N_{MC}	HMCa	HMCb	AMCa	AMCb
0.02	2.77	22.95	10000	0.13	0.25	0.36	0.48
0.01	6.15	48.17	50000	0.66	1.35	1.11	2.48
0.005	12.12	99.19	100000	1.37	2.56	1.82	4.99
0.0025	27.61	204.88	200000	2.56	5.08	3.70	9.96

Table 10: *Bates-Hull-White model. Computational times (in seconds) for European call options in Table 8 for $S_0 = 100$, $\rho_{Sr} = -0.5$.*

Δy	HTFDa	HTDFb	N_{MC}	HMCLSa	HMCLSb	AMCLSa	AMCLSb
0.02	2.77	23.10	10000	0.28	0.43	0.40	0.62
0.01	6.39	48.65	50000	0.80	1.79	1.30	2.72
0.005	12.50	99.85	100000	1.91	3.89	3.02	6.15
0.0025	27.92	205.60	200000	4.03	8.11	5.20	10.75

Table 11: *Bates-Hull-White model. Computational times (in seconds) for American call options in Table 9 for $S_0 = 100$, $\rho_{Sr} = -0.5$.*

7 Conclusions

In this paper we have studied the numerical stability of the hybrid tree/finite-difference method already introduced in [8, 9]. We have extended the method to the Bates model with a possible stochastic interest rate. We have also considered a Monte Carlo approach. We have used our numerical schemes to evaluate European and American options. The results can be considered good and reliable, and the comparison with existing pricing methods has shown a good efficiency also in terms of computing time costs.

Acknowledgements. The authors wish to thank Andrea Molent for having implemented the Alfonsi simulation scheme and the Monte Carlo Longstaff-Schwartz algorithms.

References

- [1] E. AKYILDIRIM, Y. DOLINSKY, H.M. SONER, Approximating stochastic volatility by recombining trees, *Ann. Appl. Probab.*, **24** (2014), pp. 2176–2205.
- [2] A. ALFONSI (2010): High order discretization schemes for the CIR process: application to affine term structure and Heston models, *Mathematics of Computation*, **79**, 209–237.
- [3] L.B.G. ANDERSEN (2008) Simple and efficient simulation of the Heston stochastic volatility model. *J. Comput. Financ.*, **11**, 1–42.
- [4] L.B.G. ANDERSEN, V.V. PITERBARG (2007): Moment explosions in stochastic volatility models. *Finance and Stochastics*, **11**, 29–50.
- [5] E. APPOLLONI, L. CARAMELLINO, A. ZANETTE (2015): A robust tree method for pricing American options with CIR stochastic interest rate. *IMA Journal of Management Mathematics*, **26**, 345–375.
- [6] L.V. BALLESTRA, L. CECERE (2016) A fast numerical method to price American options under the Bates model. *Computers and Mathematics with Applications*, **72**, 1305–1319.
- [7] D.S. BATES (1996): Jumps and stochastic volatility: exchange rate processes implicit in Deutsch mark options. *Rev Fin.*, **9**, 69–107
- [8] M. BRIANI, L. CARAMELLINO, A. ZANETTE (2017): A hybrid approach for the implementation of the Heston model. *IMA Journal of Management Mathematics*, **28**, 467–500.
- [9] M. BRIANI, L. CARAMELLINO, A. ZANETTE (2017): A hybrid tree/finite-difference approach for Heston-Hull-White type models. *The Journal of Computational Finance*, **21**, 1–45.
- [10] M. BRIANI, L. CARAMELLINO, G. TERENCE (2018): Convergence rate of Markov chains and hybrid numerical schemes to jump-diffusions with application to the Bates model. [ArXiv:1809.10545](https://arxiv.org/abs/1809.10545)
- [11] M. BRIANI, C. LA CHIOMA, R. NATALINI (2004): Convergence of numerical schemes for viscosity solutions to integro-differential degenerate parabolic problems arising in financial theory. *Numer. Math.*, **98**(4), 607–646.
- [12] M. BRIANI, R. NATALINI, G. RUSSO (2007): Implicit-Explicit Numerical Schemes for Jump-Diffusion Processes. *Calcolo*, **44**, 33–57.
- [13] D. BRIGO, F. MERCURIO (2006): *Interest Rate Models-Theory and Practice*. Springer, Berlin.
- [14] L. BRUGNANO, D. TRIGIANTE (1992): Tridiagonal matrices: Invertibility and conditioning, *Linear Algebra and its Applications*, **166**, 131–150.

- [15] P. CARR, D. MADAN (1999): Option valuation using the Fast Fourier Transform. *The Journal of Computational Finance*, **3**, 463-520.
- [16] C. CHIARELLA, B. KANG, G. MEYER, A. ZIOGAS (2009): The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines. *Int. J. Theor. Appl. Finan.*, **12**, 393.
- [17] C. CHIARELLA, B.KANG, G.H. MEYER (2012): The evaluation of barrier option prices under stochastic volatility. *Computers and Mathematics with Applications*, **64**, 2034-2048.
- [18] J.C. COX, J. INGERSOLL, S. ROSS (1985): A theory of the term structure of interest rates, *Econometrica*, **53**, 385-407.
- [19] R. CONT, E. VOLTCHKOVA (2005): A finite difference scheme for option pricing in jump-diffusion and exponential Lévy models. *SIAM Journal on Numerical Analysis*, **43**(4), 1596–1626.
- [20] V. D’HALLUIN, P.A. FORSYTH, G. LABAHN (2005): A semi-Lagrangian Approach for American Asian options under jump-diffusion, *Siam J.Sci.Comp.* **27**, 315-345.
- [21] D.J. DUFFY (2006). *Finite difference methods in financial engineering. A partial differential equation approach*. Wiley Finance Series.
- [22] E. EKSTRÖM, J. TYSK (2010) The Black-Scholes equation in stochastic volatility models. *J. Math. Anal. Appl.* **368**, 498–507
- [23] L. GOUDENEGE, A. MOLENT, A. ZANETTE (2016). Pricing and Hedging GLWB in the Heston and in the Black-Scholes with Stochastic Interest Rate Models. *Insurance: Mathematics and Economics* **70**, 38-57.
- [24] A.L. GRZELAK, C.W. OOSTERLEE (2011): On the Heston model with stochastic interest rates. *SIAM J. Fin. Math.* **2**, 255-286.
- [25] T. HAENTJENS, K.J. IN’T HOUT (2012): Alternating direction implicit finite difference schemes for the Heston-Hull-White partial differential equation. *J. Comp. Finan.* **16**, 83–110.
- [26] S.L. HESTON (1993): A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, *Review of Financial Studies*, **6**, 327-343.
- [27] J. HULL, A. WHITE (1994): Numerical procedures for implementing term structure models I. *Journal of Derivatives* **2**(1), 7-16.
- [28] A. ITKIN (2016): Efficient Solution of Backward Jump-Diffusion PIDEs with Splitting and Matrix Exponentials. *The Journal of Computational Finance*, **19**, 29-70.
- [29] C. KAHL, P. JÄCKEL (2006) Fast strong approximation Monte Carlo schemes for stochastic volatility models. *Quant. Finance* **6**, 513–536.
- [30] S.G. KOU (2002): A Jump-Diffusion Model for Option Pricing. *Management Science*, **48**, 1086-1101.
- [31] D. LAMBERTON, B. LAPEYRE (2008): Introduction to stochastic calculus applied to finance. Second edition. Chapman & Hall/CRC Financial Mathematics Series.
- [32] C.C. LO, D. NGUYEN, K. SKINDILIAS (2017) A Unified Tree approach for options pricing under stochastic volatility models. *Finance Research Letters* **20**, 260–268.
- [33] F.A. LONGSTAFF, E.S. SCHWARTZ (2001): Valuing American options by simulations: a simple least squares approach. *The Review of Financial Studies*, **14**, 113-148.
- [34] R. LORD, R. KOEKKOEK AND D.V. DIJK (2010) A comparison of biased simulation schemes for stochastic volatility models. *Quant. Finance* **10**, 177–194.

- [35] R.C. MERTON (1976): Option pricing when underlying stock returns are discontinuous. *J. Financial Econom.*, **3**, 125–144.
- [36] D.B. NELSON, K. RAMASWAMY (1990): Simple binomial processes as diffusion approximations in financial models. *The Review of Financial Studies*, **3**, 393–430.
- [37] PREMIA: An Option Pricer. <http://www.premia.fr>
- [38] S. SALMI, J. TOIVANEN (2014): IMEX schemes for pricing options under jump-diffusion models. *Applied Numerical Mathematics*, **84**, 33–45.
- [39] J. TOIVANEN (2010): A Componentwise Splitting Method for Pricing American Options Under the Bates Model. *Applied and Numerical Partial Differential Equations, Computational Methods in Applied Sciences*, **15**, 213–227.
- [40] M. VELLEKOOP, H. NIEUWENHUIS (2009): A tree-based method to price American Options in the Heston Model. *The Journal of Computational Finance*, **13**, 1–21.
- [41] O. VASICEK (1977): An equilibrium characterization of the term structure. *Journal of Financial Economics*, **5**, 177–188.
- [42] VOLTCHKOVA, E. AND TANKOV, P. (2008) Deterministic methods for option pricing in exponential Lévy models. PREMIA documentation. Available online at: <http://www.premia.fr>